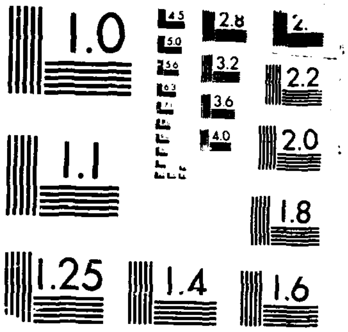
AD-A194 622 TRACKING AND CONTROL OF A NEUTRAL PARTICLE BEAM USING 1/3
MULTIPLE MODEL ADAP..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI..
UNCLASSIFIED L J HARAMBASIC DEC 87 AFIT/GE/ENG/87D-23 F/G 20/7 NL



MICROCOPY RESOLUTION TEST CHART

BUREAU OF STANDARDS-1963-A

AD-A194 622

0

DTIC FILE COPY

TRACKING AND CONTROL OF A NEUTRAL PARTICLE BEAM USING MULTIPLE MODEL ADAPTIVE MEER FILTER

THESIS

Louis Jasper Harambasic Jr., B.S.E.
Captain, USAF
AFIT/GE/ENG/87D-23

DTIC
S ELECTE D
JUN 2 3 1988
E

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved for public release and sale; its distribution is unlimited.

88 6 23 030

①

AFIT/GE/ENG/87D-23

TRACKING AND CONTROL OF A NEUTRAL PARTICLE BEAM USING MULTIPLE MODEL ADAPTIVE MEER FILTER

THESIS

Louis Jasper Harambasic Jr., B.S.E.
Captain, USAF
AFIT/GE/ENG/87D-23

DTIC
S ELECTE D
JUN 2 3 1988
E

Approved for public release; distribution unlimited

AFIT/GE/ENG/87D-23

TRACKING AND CONTROL
OF A NEUTRAL PARTICLE BEAM
USING MULTIPLE MODEL ADAPTIVE
MEER FILTER

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Louis Jasper Harambasic Jr., B.S.E.
Captain, USAF

December 1987

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Approved for public release; distribution unlimited

Preface

The purpose of this research was to continue the development of a controller that uses a Poisson space-time point measurement process for feedback. The effort was motivated in part by the concurrent research being conducted at the Air Force Weapons Laboratory, Kirtland AFB, NM. The research was conducted with the primary goal of developing a neutral particle beam controller, but the issues discussed within this thesis should be valid for other application.

No effort of this magnitude is accomplished without a great deal of support. Foremost, I would like to thank God for spirital support during this effort. I would also like to thank my readers, LtCol Zdzislaw H. Lewantowicz and Capt Steve Rogers for their patience in reviewing my work. Most of all, thanks go to my advisor Dr. Peter S. Maybeck. His extraordinary patience and unbounded enthusiasm made this effort possible. His attention to detail and concern for proper communication of ideas has given me a greater respect for the written language.

-- Louis J. Harambasic Jr.

Table of Contents

Page

Preface ii

List of Figures vii

List of Tables xi

Abstract xii

I. Introduction 1

1.2 The Problem 4

1.3 Scope 5

1.4 Approach 6

1.5 Summary of Remaining Chapter . . . 7

II. Beam Tracking 9

2.1 NPB Model 9

2.2 Beam Measurement Model 15

2.2.1 The Signal 15

2.2.2 The Noise Model 20

2.3 Synder-Fishman Filter 23

2.4 Multiple Model Adaptive Estimation . 29

2.5 Meer Filter 35

2.5.1 Basic Meer Filter. 35

2.5.2 Pruning the Hypothesis Tree . 40

2.5.2.1 Best Half Method . . 40

2.5.2.2 The Merge Method . . 44

2.6 MMAE with Meer Elemental Filters . . 48

2.7 Summary 50

III. Controller Design 54

3.1 Target Model and Filter 56

3.1.1 Target Truth Model . . . 57

3.1.2 Target Filter 60

3.2 Controller Gain Via LQ Synthesis . . 68

3.2.1 Description of Quadratic Cost Function 68

3.2.2 Controller Gain Synthesis . . 75

3.2.3 Weighting Matrices using Implicit Model Following 78

3.2.3.1 Explicit Model Following 79

3.2.3.2 Implicit Model Following 79

3.3 Controller Structure 86

3.3.1 Proportional Gain Regulator . 86

3.3.2 Proportional Gain Tracker . . 87

3.3.3 Proportional Plus Integral Regulator 90

3.3.4 Proportional Plus Integral Tracker 93

3.3.5 Multiple Model Adaptive Controller 94

3.3.6 Choosing a Controller . . 96

3.4 Summary 100

IV. Simulation Design and Evaluation Procedure . 102

4.1 Performance Analysis 103

4.1.1 Evaluating the MMAE Meer Filter . 104

4.1.2 Evaluating the MMAC with Implicit Model Following Weighting Matrices 107

4.1.3 The Method - Monte Carlo Simulation 110

4.2 The Tools - SOFE and SOFEPL . . . 111

4.2.1 Data Gathering with SOFE . . 111
4.2.1.1 Random Interval Updating 112
4.2.1.2 Truth Models . . . 115
4.2.1.3 Target Kalman Filter . 116
4.2.1.4 Meer Filter . . . 117
4.2.1.5 Controller Simulation . 119
4.2.1.6 Additional Alterations . 119
4.2.2 Assimilating the Data - SOFEPL . 120
4.2.2.1 MMAE Meer Statistics . 120
4.2.2.2 Controller Statistics . 123
4.3 Filter, Controller, and Simulation Parameters 124
4.3.1 The Beam and Target Parameters . 125
4.3.2 Nominal Parameters . . . 127
4.3.3 Analysis Parameters . . . 129
4.4 Summary 130
V. Results and Analysis 132
5.1 MMAE Meer Filter State Estimation Performance 133
5.2 MMAE Meer Filter Parameter Estimation Performance 140
5.3 Simplified MMAE Meer Filter State Estimation Performance 148
5.4 MMAE Meer Performance with Variable Beam Time Constant 151
5.5 MMAE Meer Filter Range with Three Meer Filters 154
5.6 MMAE Meer Filter in Controller Loop . 161
5.6.1 Beam Following Detector . . 161

5.6.2 Divergence Detection and Filter Reset 177
5.6.3 MMAE Meer Baseline 179
5.6.4 Controller Baseline 202
5.7 Implicit Model Following Controller Performance 207
5.8 Summary 207
VI. Conclusions and Recommendations 210
6.1 Conclusions 210
6.2 Recommendations 213
Vita 223

List of Figures

Page

1.1 NPB Concept 2

2.1 Transitions for the Hydrogen Atom 14

2.2 Radiation Intensity versus Emission Direction 16

2.3 Signal Arrival Rate 19

2.4 Signal and Noise Rate Functions . . . 24

2.5 First Order Gauss-Markov Process . . . 28

2.6 Multiple Model Filtering Algorithm . . 34

2.7 Conditional Probability Tree 38

2.8 Meer Filter Structure 41

2.9a Best Half Pruning with Signal Most Probable . 43

2.9b Best Half Pruning with Noise Most Probable . 43

2.10 Merge Method Hypothesis Merging . . . 45

2.11 Merge Method Pruning 46

2.12 MMAE Meer State Estimation 51

2.13 MMAE Meer Parameter Estimation . . . 52

3.1 Controller Structure 55

3.2 Target Model 59

3.3 Target Dynamics and Sensor Models 64

3.4 Target Kalman Filter Block Diagram . . 66

3.5a Pseudo-Integral for a Regulator . . . 92

3.5b Pseudo-Integral for a Tracker . . . 92

3.6 Elemental PI Controller Block Diagram . . 95

3.7 Multiple Model Adaptive Controller . . 97

4.1 Macro Level Flow of Modified SOFE . . . 121

5.1 MMAE Meer State Estimate Mean Error Standard Deviation; True Tau = 20.4 . . 136

5.2 Meer Filter #1 Mean Residual 139

5.3 Adaptive Beam Time Constant True Tau = 20.4 143

5.4 Probability History of Meer Filter #1 Filter Tau = 20.0 144

5.5 Probability History of Meer Filter #2 Filter Tau = 20.8 145

5.6 Probability History of Meer Filter #3 Filter Tau = 19.2 146

5.7 Adaptive Beam Time Constant Extended Run 147

5.8 Adaptive Beam Time Constant; Downramping Tau True Tau = 20.8 - .016t 153

5.9 Adaptive State Estimation Error True Tau = 70 155

5.10 Meer Filter #3 State Estimation Error Filter Tau = 2 True Tau = 70 . . . 156

5.11 Mean Adaptive Beam Time Constant True Tau = 70 160

5.12 MMAE State Estimation Error; NPB Rotation True Tau = 25 164

5.13 Meer Filter #3 State Estimation Error True Tau = 25 Filter Tau = 10 . . . 165

5.14 Tracking Error for NPB Rotation . . . 168

5.15 MMAE State Estimation Error Detector Follows Beam 170

5.16 Meer Filter #2 State Estimation Error Detector Follows Beam 171

5.17 Propagation Error versus Beam Position . . 174

5.18 Mean Residual Meer Filter #3 Filter Tau = 19.6 True Tau = 20.2 . . 176

5.19 MMAE State Estimation Error; Oscillating True Tau = 20 + .8 sin(0.063 t) 181

5.20 Meer Filter #1 State Estimation Error
Filter Tau = 20; Oscillating True Tau . . 182

5.21 Adaptive Beam Time Constant; Oscillating True
Tau = 20 + .8 sin(0.063 t) 184

5.22 MMAE State Estimation Error
True Tau = 20.4 185

5.23 Meer Filter #2 State Estimation Error
Filter Tau = 20.8 True Tau = 20.4 . . 186

5.24 Closed Loop Mean Adaptive Beam Time Constant
True Tau = 20.4 187

5.25 Closed Loop Probability History Meer Filter #1
Filter Tau = 20.0 True Tau = 20.4 . . 188

5.26 Closed Loop Probability History Meer Filter #2
Filter Tau = 20.8 True Tau = 20.4 . . 189

5.27 Closed Loop Probability History Meer Filter #3
Filter Tau = 19.6 True Tau = 20.4 . . 190

5.28 Closed Loop MMAE State Estimation Error
Mean +/- One Standard Deviation
True Tau = 20.0 192

5.29 Closed Loop Meer #1 State Estimation Error
Filter Tau = 20.0 True Tau = 20.0 . . 193

5.30 Closed Loop RMS Beam Error Meer #1
Filter Tau = 20.0 True Tau = 20.0 . . 194

5.31 Closed Loop Mean Adaptive Beam Time Constant
True Tau = 20.0 195

5.32 Closed Loop Mean Probability Meer Filter #1 .
Filter Tau =20.0 True Tau = 20.0 . . . 197

5.33 Closed Loop Mean Probability Meer Filter #2 .
Filter Tau =20.8 True Tau = 20.0 . . . 198

5.34 Closed Loop Mean Probability Meer Filter #3 .
Filter Tau =19.2 True Tau = 20.0 . . . 199

5.35 Meer Filter Without Measurements
Filter Tau =20.0 True Tau = 20.0 . . . 200

5.36 Meer Filter With Measurements
Filter Tau =20.0 True Tau = 20.0 . . . 201

5.37 Controller RMS Tracking Error
Using Full-State Feedback 204

5.37 Controller RMS Tracking Error Using Single Filter Estimates 205

5.38 Nine Element MMAC RMS Tracking Error . . . 206

List of Tables

Page

2.1 Hypothesis Sequences 36

5.1 MMAE Meer State Estimation Performance . . 135

5.2 Individual Meer State Estimation Performance 138

5.3 Meer Filter Pseudo-Residual Performance . 138

5.4 MMAE Meer Parameter Estimate Performance . 141

5.5 State Estimate Performance and Computational Load for Depths of 1 and 3 149

5.6 State Estimate Performance With Variable Beam Time Constant 152

5.7 Stressed Filter Performance 157

5.8 Filter Performance with Detector Following Beam 172

5.9 Controller Baselines 203

AFIT/GE/ENG/87D-23

Abstract

The purpose is to point the centroid of a Neutral Particle Beam (NPB) at an intended target. A Multiple Model Adaptive Estimator using elemental Meer Filters is used to estimate the centroid of a NPB model as a one-dimensional first-order Gauss-Markov position process. The MMAE Meer Filter is also used to estimate the beam time constant. "Merge Method" of filter pruning is used to limit the size of the elemental Meer filters. A bank of three Kalman filters are used to estimate the states of the target which has a variable dynamics driving noise strength. The target is modelled as a third-order Gauss-Markov position process. A Multiple Model Adaptive Controller is designed using LQG methods, and true states are replaced by their best estimates by invoking the principle of assumed certainty equivalance. MMAE Meer Filter performance analysis is performed for an uncontrolled beam and for a controlled beam. Controller baselines are established.

I. Introduction

This study is motivated by the research on Neutral Particle Beams (NPB) being done at the Air Force Weapons Laboratory as a part of the Strategic Defense Initiative. Most previous interest in the Neutral Particle Beam has been its use as a weapon. Figure 1.1 is one concept of a neutral particle beam weapon. Although there remains a strong interest in NPB weapons, the Air Force's current primary interest in the NPB is to use it to determine the material components of a target [14]. This is to be accomplished through the nuclear interactions of the particle beam with the target. As it is necessary to direct the beam onto the target to perform this mission, we must be able to control the beam. Prior to being able to control the beam, we must be able to estimate the beam's location. Thus, the purpose of this study is to continue the research on tracking and control of the NPB.

One approach to estimating the beam location involves injecting laser energy into the beam. Photons absorbed by the beam and then reemitted are collected on a detector array. The pattern of the photon arrivals is used to infer information about the beam's location. The photon events can be modelled as Poisson space-time point processes. Snyder and Fishman [29] developed an estimator to incorporate such events to estimate the centroid of the

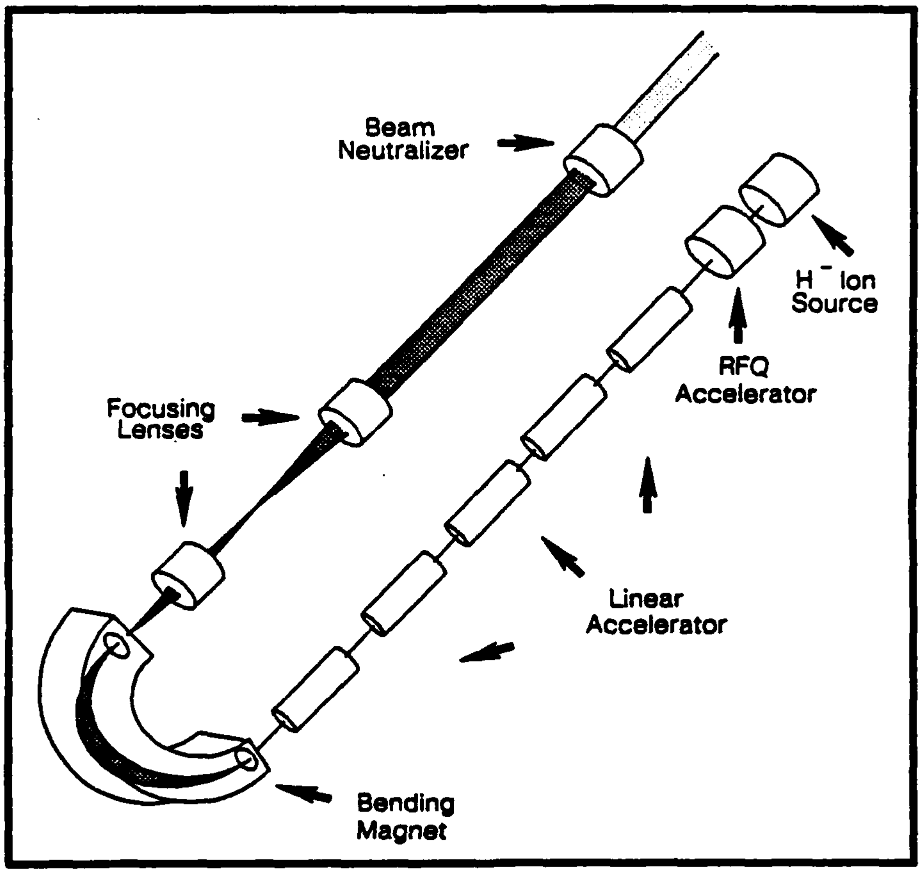


Figure 1.1 NPB Concept [20]

source of photon events. The structure of the Snyder Fishman (SF) filter is similar to that of the Kalman filter structure. The significant difference is that the arrival times of events in the SF filter are not known a priori. Santiago [27] showed that the SF filter is very sensitive to noise induced point process events that are representative of "dark currents" found in photon detectors. In 1982, David Meer in his doctoral dissertation [21] developed an estimator that filtered out the noise events. The Meer filter is based on a bank of SF filters. Each SF filter in the bank assumes a different sequence of signal-induced or noise-induced events. Those SF filters which assume the current event is a signal, use the event to update the filter's estimate. Those SF filters which assume the current event is a noise, ignore the event. As there is an ever-growing number of sequences of signal and noise events, this requires an ever-growing number of SF filters. Meer then devised a method to limit the number of SF filters through the "Best Half" method. This method of limiting the number of SF filters was replaced by the "Merge" method incorporated by Zicker in 1983 [32].

Once the beam estimation problem had been solved, the problem of beam control was examined. Zicker conducted a feasiblity study with a proportional gain controller [32]. This was followed by the design of proportional plus integral (PI) controllers by Moose and Jamerson [8,23]. The type one control provided by the PI controller resulted in superior tracking performance. Jamerson also incorporated a more realistic target model in his study. This target model was based on a third order Gauss-Markov position process, rather than a first order Gauss-Markov position process as in the previous studies. PI controller performance against the new target was found to be sluggish. Johnson then designed a multiple model adaptive controller with elemental PI controllers based on separate assumptions about the strength of the target dynamics driving noise. Each of these controllers was based on Linear models and Quadratic cost with full-state feedback to allow LQ synthesis techniques [18] to be used in the determination of constant controller gains. Then by applying the principle of assumed certainty equivalence [18], the full-state feedback was replaced by estimates provided by the Meer beam filter and Kalman target filter.

1.2 The Problem

In examining the robustness of the PI controller to variations in true beam parameters (while filter assumed values remained constant), Johnson uncovered a significant robustness problem when the beam time constant was mismodelled. For small variations in the true beam time constant, the Meer filter was unable to estimate the beam position precisely and the controller went unstable. Since stability robustness is the heart of any feedback controller design, this problem had to be evaluated and a solution found. Johnson evaluated the problem and one of his recommendations was to apply multiple model adaptive estimation to the beam time constant [9].

1.3 Scope

This document assumes the reader is familiar with the basic elements of control theory and stochastic processes. The beam, target and measurement models of this study are limited to one spatial dimension. This is done to allow simple performance characteristics to be studied without the complications of cross correlation between multiple states and parameters. The target and beam are assumed to be well described by linear models. Where reasonable, random variables or distributions are assumed to be, or simplified to be, either Gaussian or uniform as appropriate. Controller design is based on LQ synthesis and Multiple Model Adaptive Controller techniques. Implicit Model Following techniques are investigated as a method to determine controller gains which offer robustness and performance. Beam actuators are assumed to be linear devices commanded through zero-order-hold circuits that maintain the commanded feedback until the following controller update.

1.4 Approach

This project begins by designing a multiple model adaptive estimator using elemental Meer filters based on separate assumptions of the beam time constant. The weighted state estimates are combined to provide an adaptive state estimate. The same weightings are applied to the filter-assumed beam time constants to provide an adaptive parameter estimate. The weightings are based on the residual performance of the individual filters. The controller uses Multiple Model Adaptive Controller techniques to design an adaptive controller based on a bank of PI controllers. Each elemental controller is based on a separate combination of assumed beam time constant and strength of the target dynamics driving noise. In an alternative approach, the parameter estimate from the MMAE Meer is used to propagate the state estimate of a separate Meer filter that is updated with the adaptive state estimate. This adaptive estimate is then supplied to a controller based on a nominal beam time constant.

As in previous controller designs, the constant controller gains are determined through LQ synthesis techniques assuming full-state feedback. However, in this study, Implicit Model Following is used as an aid to determine the set of weighting matrices that results in a controller that provides robustness and performance. The full-state feedback is then replaced by the Meer and Kalman filter estimates through the application of assumed certainty equivalence [18].

The estimator and controller designs are validated using computer simulations. Since any adaptive filter is nonlinear and the resulting filter-controller combination is also nonlinear, covariance analysis is inappropriate for this simulation. Instead, Monte Carlo simulations are used to collect statistics to evaluate performance. Controller performance is measured against two baselines. The first baseline is a controller with full-state feedback and access to the true parameter values. The second baseline is a controller receiving estimates from a single beam and target filter that have access to the true parameters. The first baseline indicates the best possible performance and robustness for a given controller design. The second baseline provides a realistic assessment of the best performance possible from controllers that operate with state estimates rather than the actual states.

1.5 Summary of Remaining Chapters

Chapter II develops the estimator. The chapter includes information on the the beam model, space time point processes, Multiple Model Adaptive Estimation, the Meer filter, hypothesis pruning, and the MMAE Meer filter. Chapter III develops the controller. Information in this chapter includes the target model, LQ synthesis, Implicit

Model Following, and various controller structures. Chapter IV describes the performance analysis of the estimator and controller. This includes the analyses to be performed and the software tools used to generate and collect the data. Chapter V contains the results of the analyses on the MMAE Meer filter and the controllers. Time constraints prevented the analysis of a controller with an Implicit Model Following based design. Chapter VI contains the conclusions of this study and recommendations for subsequent studies.

II. Beam Tracking

The first step in tracking and pointing of a Neutral Particle Beam (NPB) is the determination of the beam's location, or tracking the beam. In his doctoral effort, Meer [21] developed a Multiple Model Adaptive Estimator (MMAE) using elemental Snyder-Fishman filters to determine the location of the beam centroid. This chapter will first describe the NPB dynamics model. Next, it will cover the space-time point process model and how it applies to the case of measurements that can be extracted from a NPB being irradiated with laser beams. Then, the Snyder-Fishman filter which uses that space-time point process as its assumed measurement model will be described. The Multiple Model Adaptive Estimator will be presented to describe the basic structure of the Meer and MMAE Meer filters. Then, the Meer filter will be described which uses a bank of Snyder-Fishman filters to estimate the NPB centroid in the presence of noise measurement events. Tree pruning and merging will then be described as methods of making the Meer filter computationally realizable. Finally, a MMAE using elemental Meer filters will be developed as a way to accommodate the effects of an uncertain beam time constant.

2.1 NPB Model

As its name implies, the NPB is a beam of particles. In modeling the NPB, we are interested in both its nature as individual particles and as a beam. This section begins with a description of the particle aspect of the NPB. This will be followed by a description of the beam nature that is of interest to us.

The term neutral particle indicates the NPB is a collection of electrically neutral particles, for instance, monoatomic hydrogen, which is composed of a single negatively charged electron orbiting a single positively charged proton. This gives the NPB its electrically neutral characteristics while propagating in space [20]. One approach to modelling the NPB might be to model the dynamics of individual particles through the equation [20:Chap 1,4-5]

$$\underline{f} = \gamma m_0 \, (d\underline{v}/dt + \gamma^2 \, v/c^2 \, dv/dt \, \underline{v}) \qquad (2\text{-}1)$$

where

$\underline{f}$ is the force vector

m_0 is the rest mass of the particle

$\underline{v}$ is the velocity vector

v is the scalar magnitude of the velocity

c is the speed of light

γ is a relativistic term given by

$$\gamma = [\, 1 - (v/c)^2 \,]^{-1/2} \qquad (2\text{-}2)$$

Although this approach might be highly accurate, to model a lethal NPB would require on the order of 10^{21} such models [25:59]. Fortunately, McKee indicates that for the portion of the NPB that we are interested in, there are essentially no collisions between particles [20:8-1]. Since there are no collisions, a group of these particles traveling in essentially the same direction can be viewed as traveling along a long tube or cylinder, in other words, viewed as a beam. The centerline of this cylinder describes the centroid of the beam. The centerline also describes the direction of the beam velocity vector.

Since the objective is to point the beam, or to align the velocity vector with the target, the movement of the beam centroid needs to be modelled. To model the motion of the centroid, we must consider its expected response to our control inputs and additionally to disturbance inputs and dynamics driving noises. For this study we will consider the motion of the beam centroid to be adequately modeled by the linear stochastic differential equation [16:204]:

$$\underline{dx}(t) = \underline{F}(t)\underline{x}(t)dt + \underline{B}(t)\underline{u}(t)dt + \underline{G}(t)\underline{d\beta}(t) \qquad \text{(2-3a)}$$

or, in the less rigorous white noise notation [16:204]:

$$d\underline{x}(t)/dt = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \qquad \text{(2-3b)}$$

where

$\underline{x}(t)$ is a stochastic vector of states of dimension n characterizing the motion of the beam centroid. $\underline{H}\ \underline{x}$ is the beam centroid location, as indicated on the detector array ($\underline{H}\ \underline{x}$ is a projection from R^n onto R^m; m=2 for a plane, m=1 for this experiment)

$\underline{F}(t)$ is an n-by-n matrix describing the system dynamics

$\underline{B}(t)$ is an n-by-r deterministic input matrix relating the control inputs to the beam states

$\underline{u}(t)$ is an r-vector of piecewise constant deterministic control input functions

$\underline{G}(t)$ is an n-by-s noise input matrix relating the dynamic driving noises to the beam states

$\underline{\beta}(t)$ is an s-vector Brownian motion with statistics (E is the expectation):

$$\begin{gathered} E\left\{\underline{\beta}(t)\right\} = \underline{0} \\ E\left\{[\underline{\beta}(t)-\underline{\beta}(t')][\underline{\beta}(t)-\underline{\beta}(t')]^T\right\} = \int_t^{t'} \underline{Q}(\alpha)d\alpha \end{gathered} \tag{2-4a}$$

with $\underline{Q}$ an s-by-s matrix of piecewise continuous functions such that $\underline{Q}(t)$ is symmetric and positive semidefinite.

$\underline{w}(t)$ is a s-vector white Gaussian noise (the hypothetical derivative of $\underline{\beta}$ above) with statistics:

$$\begin{gathered} E\left\{\underline{w}(t)\right\} = \underline{0} \\ E\left\{\underline{w}(t)\underline{w}(t')^T\right\} = \underline{Q}(t)\delta(t-t') \end{gathered} \tag{2-4b}$$

with the same description of $\underline{Q}$ just given [16:204]

The beam centroid states are propagated using Equation (2-3) from the initial condition:

$$\underline{x}(t_0) = \underline{x}_0 \qquad (2\text{-}5)$$

where $\underline{x}_0$ has statistics given by:

$$E\{\underline{x}(t_0)\} = \underline{m}_0 \qquad (2\text{-}6a)$$

$$E\{[\underline{x}(t_0)-\underline{m}_0][\underline{x}(t_0)-\underline{m}_0]^T\} = \underline{P}_0 \qquad (2\text{-}6b)$$

where:

$\underline{m}_0$ is the mean state initial condition

$\underline{P}_0$ is an n-by-n covariance matrix that is symmetric and positive semidefinite

For pointing of the NPB, the states we need to know are the location and the *direction* of the velocity vector or, alternatively, the centroid location at two points along the "cylinder" at essentially the same time. Either case requires one to gather some information on particle locations in the beam. One approach to gathering this information is to shine a laser beam into the particle beam to "excite" the electrons in the beam into higher energy levels or quantum states [26]. When the beam electrons "relax" they release this energy in the form of photons at specific quantum levels [13:16-18] (See Figure 2.1). These photons can then be collected by using a photodetector array. The number of photons that arrive in a given amount of time, at a given location of the array is related to the

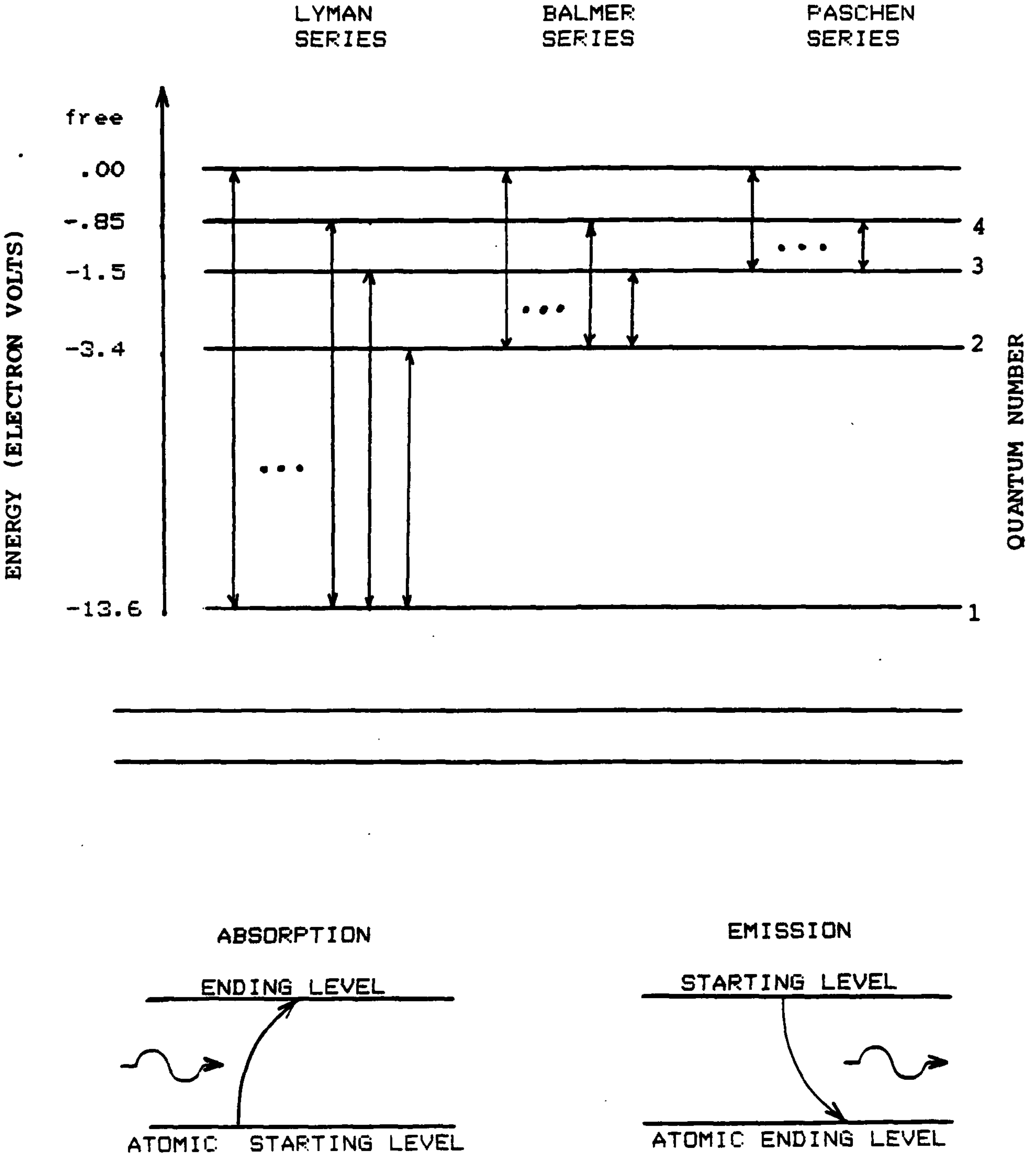


Figure 2.1 Transitions for the Hydrogen Atom[13]

location of the electrons releasing the photons. An electron in the detector that is energized by a photon provides a signal that a photon event has occurred. The model for the location and time of arrival of these photon events is covered in the next section.

2.2 Beam Measurement Model

This section models the signal and noise processes occurring at the detector array. Individual signal and noise events are described as point processes. The signal is described as a Poisson space-time point process with an arrival rate density that has a Gaussian spatial distribution. Included in the signal model is an array description with a method for determining appropriate array size. Then, we will consider the presence of noise at the array. The noise is modeled as a space-time point process uniformly distributed over the detector. This section concludes with expressions for combining signal and noise process.

2.2.1 The Signal. A thorough description of the emission of photons from a NPB and their detection at an array is not necessary for this study. Therefore, the objective of this section is to provide only enough information to justify the selected model. To model the signal, we would like to know the rate of the input signal. Figure 2.2 depicts the emission pattern for relativistic

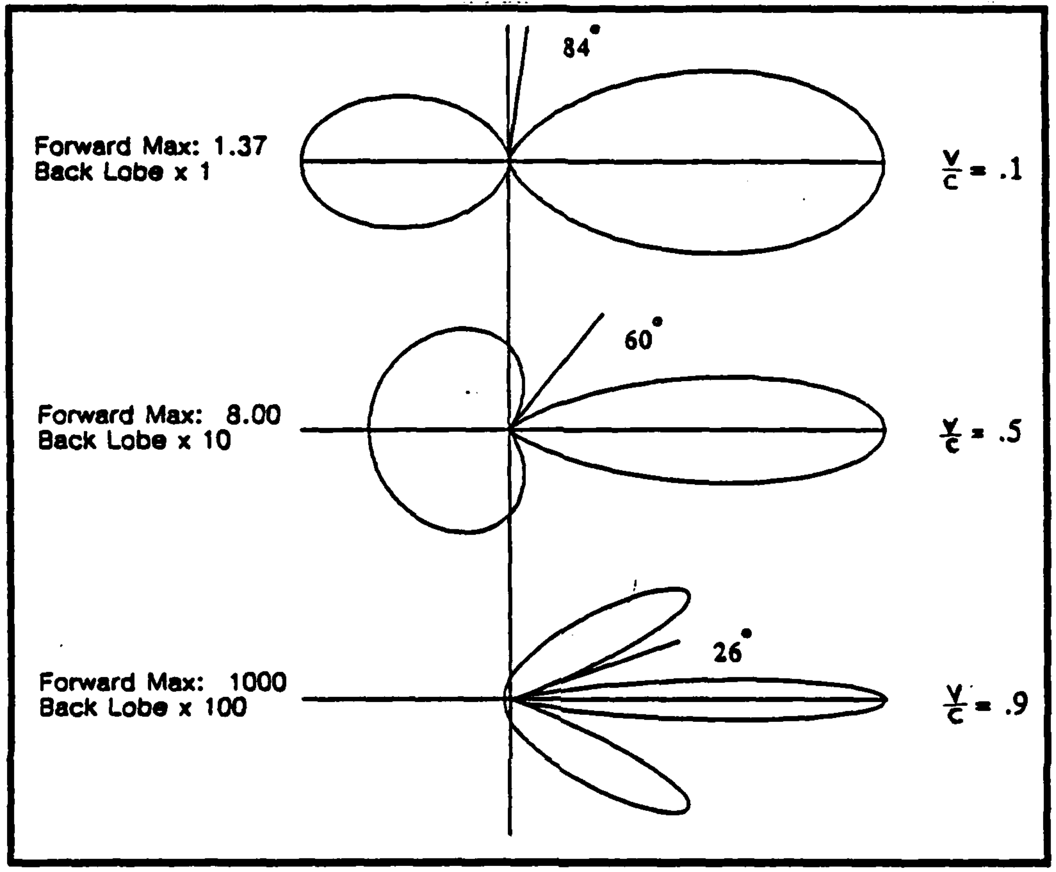


Figure 2.2 Radiation Intensity versus Emission Direction for a Unit Charge at Various Velocities. Forward maximum is intensity multiplication relative to a non-relativistic charge. The back lobe has been multiplied by the given factor to show detail. The line labeled with an angle is the direction of zero emission. [20]

particles. The figure indicates only a small portion of the photons emitted in the NPB will arrive at a detector located to the side of the beam. As described in Section 2.1, the emission of photons from the NPB is at discrete quantum levels. These same quantum effects occur at the detector surface. As a photon interacts with the atoms in the detector, an electron may absorb the energy of the photon, thereby transitioning to a higher energy level. When a transition occurs, an event is declared with a particular time and location. Thus, there is no uncertainty in time or location of the event occurrence. Meer and Santiago [21,27,29] further described the arrival of individual photon events at the detector array as a Poisson point process. Meer [21] went on to relate this signal particle density to the projection of the beam centroid onto the detector array as (for the one dimensional case):

$$\lambda_s(t,r,x(t)) = \Gamma(t)\exp(-[r-H(t)x(t)]^T R^{-1}(t)\ [r-H(t)x(t)]/2) \tag{2-7}$$

where

λ_s is the signal rate: the arrival of photons at time t at position r on the detector, given that the state from Equation (2-3) is of value x(t) so that the true beam centroid is at H(t)x(t)

$\Gamma(t)$ is the maximum amplitude of the rate function

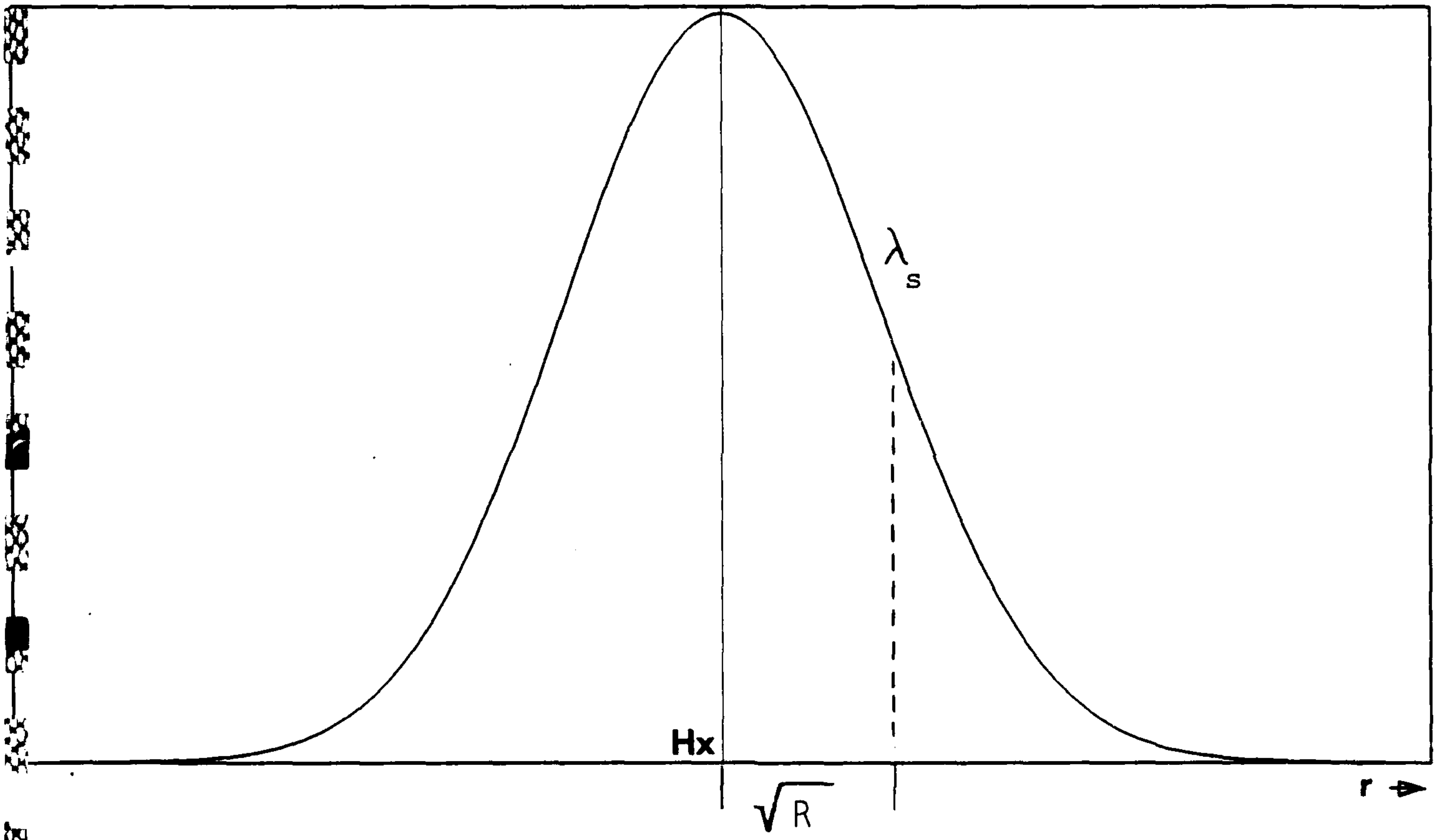
r is the spatial location of the signal event on the detector

$H(t)x(t)$ is the signal-generated beam centroid in R^1; a projection of the beam state $x(t)$ onto the line of detectors

$x(t)$ is a state stochastic process underlying the beam location in one dimensional space

R describes the spread of the beam and is therefore positive definite. The square root of R is the beam halfwidth (analogous to a standard deviation in a Gaussian probability density)

Figure 2.3 depicts the arrival rate of photons or signal particle density along a line of detectors perpendicular to the beam centerline. As can be seen from the figure, the tails of the Gaussian distribution go to infinity; therefore only an infinitely long line of detectors can collect information on the entire density function. This is physically impossible and we must limit the length of the array. If σ is the standard deviation, $\sigma = \sqrt{R}$ (the beam halfwidth) then 68.3% of the photon arrivals occur in the interval (dropping the time arguments) $[(Hx - \sigma),(Hx + \sigma)]$. Similarly 99.7% of the photons arrivals lie in the interval $[(Hx - 3\sigma),(Hx + 3\sigma)]$. Thus, it is possible to define a detector size to collect essentially all the information in the density function. Santiago [27:27-28] defines a good "rule of thumb" for determining detector size as having the detector large enough such that the distance from the centerpoint of the Gaussian rate function to the edge of the



Maximum height of λ_s curve is Γ

Total area under the curve is $\sqrt{2\pi R}\ \Gamma$

Figure 2.3 Signal Arrival Rate λ_s

detector array, is at least six beamwidths. According to Santiago [27:28] this allows the beam reasonably large wanderings while still allowing edge effects to be ignored. This results in 12 beamwidths or 24 beam halfwidths as the appropriate length for the entire line of detectors. Although we will be truncating the Gaussian distribution, we will still describe the the arrival rate as Gaussian for mathematical tractability. In reality, the array will also have to be discretized spatially to provide position information. This will limit the spatial resolution that is possible. However, for this study, the array will be assumed to be divided into infinitely small elements, and thus each photoelectron event yields both a time and exact location of the event on the detector array. Next we will consider the presence of noise in the detector and model it.

2.2.2 The Noise Model. The sources of detector noise are basically two-fold, those external to the detector and those inherent to the detector itself [13]. External noise sources will be covered briefly, followed by a description of inherent noises. A total noise model will then be described.

External or background noise is not a true type of noise, but refers to sensor output from real sources of radiation that are different from the desired source [13]. For a spaceborne NPB, some of the background noise inputs are the sun, stars, earth and man-made objects, and the like. For this study it is assumed these sources are predictable and the NPB can be designed to eliminate their impact, or that their effects can be included in the form of models to be used for inherent noise sources. Therefore, no separate model will be developed for the external noise sources.

Unfortunately, noise sources inherent in a detector cannot be eliminated and therefore must be modeled. Types of inherent detector noise for photodetectors are Generation-Recombination (G-R) noise and Dark Currents [13]. Here we have assumed that only detectors capable of detecting individual photons, such as photomultiplier tubes, avalanche photodiodes, or semiconductor photodiodes will be used as detectors. G-R noise is the normal generation of a free electron due to an incident photon which recombines within the semiconductor lattice prior to being output as a signal [13]. This results in the loss of a signal event. On the other hand, Dark Current is the output of an apparent signal event that was not generated by a photon. This is due to the statistical distribution of electron energies in the detector materials [13]. This results in a noise event.

Assuming perfectly uniform detectors, the distribution of the noise arrival rate will also be uniform. Therefore we will model the noise arrival rate or Poisson point process events as [21:181]:

$$\lambda_n(t,r) = \begin{cases} \lambda_n = \text{constant} & \text{for } -L/2 \le r \le L/2 \\ 0 & \text{for } r < -L/2,\ r > L/2 \end{cases} \qquad (2\text{-}8)$$

where λ_n is the noise rate and L is the length of the detector array.

Although the G-R noise is clearly dependent on the incoming photons, we will assume the effect of this noise can be modeled as a uniform amplitude reduction in the signal arrival rate density and need not be modeled separately. As the noise arrival rate due to Dark Current is independent of the presence of photons, we will assume the noise point process is independent of the signal point process. As Meer demonstrated, the signal point process and noise point process can be combined into a single Poisson point process with rate parameter given by [21:21-23]:

$$\lambda(t,r) = \lambda_s(t,r,x(t)) + \lambda_n(t,r) \qquad (2\text{-}9)$$

Often in signal processing that involves noise, it is desirable to consider the concept of a signal to noise ratio (SNR). For this study the SNR is defined as the average number of signal induced events produced for every noise event. The SNR, thus, can be expressed as [21]:

$$\text{SNR} = [p/(t_f - t_0)]/[\lambda_n(t,r)\ L] \qquad (2\text{-}10a)$$

$$= \sqrt{2\pi R}\ r\ /[\lambda_n(t,r)\ L] \qquad (2\text{-}10b)$$

where

p is the number of signal-induced events occurring over the period $t_f - t_0$

$\lambda_n(t,r)$ is the noise arrival rate per unit length of the detector array given in Equation (2-10)

L is the length of the detector array

Figure 2.4 gives a graphical comparison of the signal and noise models at a selected signal to noise ratio upon an appropriately sized detector. The next step is to develop an efficient method to recover the beam location information from the signal and noise information using the models we have developed.

2.3 Snyder-Fishman Filter

To determine the centerline of the beam on the array we need to use an estimator that can optimally process all the photon measurements that we provide it. It must be able to take full advantage of the Gaussian model we are assuming for spatial distribution of signal-induced photon arrival rate at the detector. In addition, it must be able to process events that arrive at random intervals. In 1975 Snyder and Fishman described a filter to "estimate the centroid of a swarm of fireflies based on their light flashes." [29]. In their work, they described the firefly flashes as Poisson point processes. Based on this, they developed a minimum mean square error estimator to determine

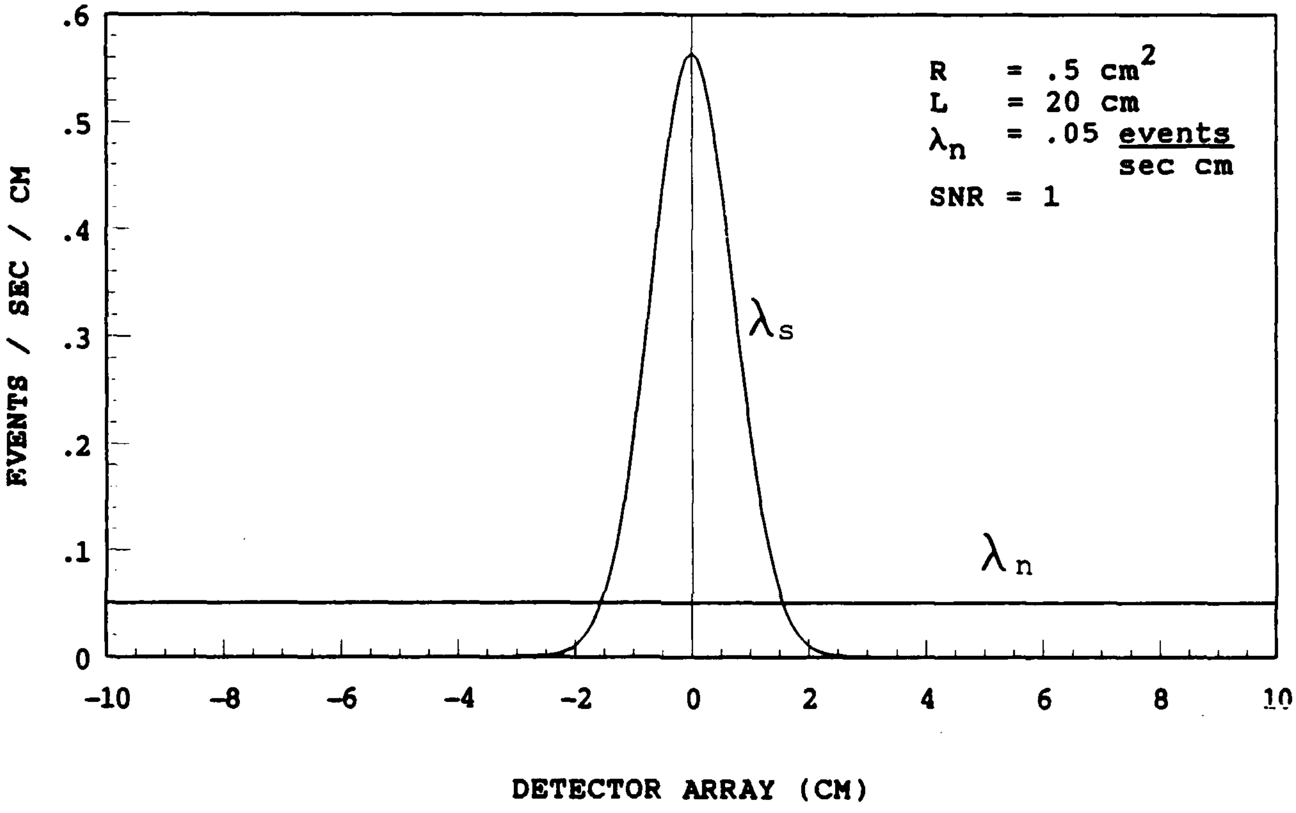


Figure 2.4 Signal and Noise Rate Functions

the centroid of the swarm. The estimator is described by the following differential equations [9:14-15,29]:

$$d\hat{\underline{x}}(t) = \underline{F}(t)\hat{\underline{x}}(t)dt + \underline{B}(t)\underline{u}(t)dt + \int_{R^m} \underline{K}(t)[\underline{r}-\underline{H}(t)\hat{\underline{x}}(t)]\; N(dt \times d\underline{r}) \qquad (2\text{-}11)$$

$$d\underline{P}(t) = \underline{F}(t)\underline{P}(t)dt + \underline{P}(t)\underline{F}^T(t)dt + \underline{G}(t)\underline{G}^T(t)dt - \int_{R^m} \underline{K}(t)\underline{H}(t)\underline{P}(t)N(dt \times d\underline{r}) \qquad (2\text{-}12)$$

$$\underline{K}(t) = \underline{P}(t)\underline{H}^T(t)[\underline{H}(t)\underline{P}(t)\underline{H}^T(t) + \underline{R}(t)]^{-1} \qquad (2\text{-}13)$$

where

$\underline{x}(t_0) = \underline{x}_0$ and $\underline{P}(t_0) = \underline{P}_0$ are the initial conditions

$\underline{P}(t)$ is the filter-computed error covariance

$\underline{G}(t)\underline{G}^T(t) = \underline{G}(t)\underline{Q}(t)\underline{G}^T(t)$; with $\underline{Q}(t)=\underline{I}$ (See Equation (2-4))

$\underline{K}(t)$ is the filter gain

The notation found in Equations (2-11) and (2-12) involves counting integrals, where:

$$\int_{R^m} f(t,\underline{r})N(dt \times d\underline{r}) = \begin{cases} 0 & N_t < 0 \\ \sum_{i=1}^{N_t} f(t_i,\underline{r}_i)\;\delta(t,t_i) & N_t \geq 1 \end{cases} \qquad (2\text{-}14)$$

and $\delta(t,t_i)$ is a Kronecker delta. Simply stated, if no events are detected, then the integral equals zero. If an event is detected, the integral causes a jump discontinuity,

$f(t,\underline{r})$, to be added to the solution of the differential equation for time t_i [29].

The filter's implementable propagation and update equations can be derived from Equations (2-11) and (2-12). The propagating equations for use between (signal-induced) events are:

$$d\underline{\hat{x}}(t) = \underline{F}(t)\underline{\hat{x}}(t)dt + \underline{B}(t)\underline{u}(t)dt \tag{2-15}$$

$$d\underline{P}(t) = \underline{F}(t)\underline{P}(t)dt + \underline{P}(t)\underline{F}^T(t)dt + \underline{G}(t)\underline{G}^T(t)dt \tag{2-16}$$

or in the form of stochastic difference equations:

$$\underline{\hat{x}}(t_{i+1}^-) = \underline{\Phi}(t_{i+1},t_i)\underline{\hat{x}}(t_i^+) + \underline{B}_d(t_i)\underline{u}(t_i) \tag{2-17}$$

$$\underline{P}(t_{i+1}^-) = \underline{\Phi}(t_{i+1},t_i)\underline{P}(t_i^+)\,\underline{\Phi}^T(t_{i+1},t_i) + \underline{Q}_d(t_i) \tag{2-18}$$

where:

$\underline{\Phi}(t_{i+1},t_i)$ is the state transition matrix associated with $\underline{F}$

$\underline{Q}_d$ is the equivalent discrete-time noise strength calculated by:

$$\underline{Q}_d(t_i) = \int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},\alpha)\underline{G}(\alpha)\underline{Q}(\alpha)\underline{G}^T(\alpha)\underline{\Phi}^T(t_{i+1},\alpha)d\alpha \tag{2-19}$$

where $\underline{Q} = \underline{I}$. When an event has been detected, a measurement update can take place as defined by the following equations:

$$\hat{\underline{x}}(t_i^+) = \hat{\underline{x}}(t_i^-) + \underline{K}(t_i)[\underline{r}_i - \underline{H}(t_i)\hat{\underline{x}}(t_i^-)] \qquad (2\text{-}20)$$

$$\underline{P}(t_i^+) = \underline{P}(t_i^-) - \underline{K}(t_i)\underline{H}(t_i)\underline{P}(t_i^-)] \qquad (2\text{-}21)$$

where the Snyder-Fishman gain $\underline{K}$ is defined by:

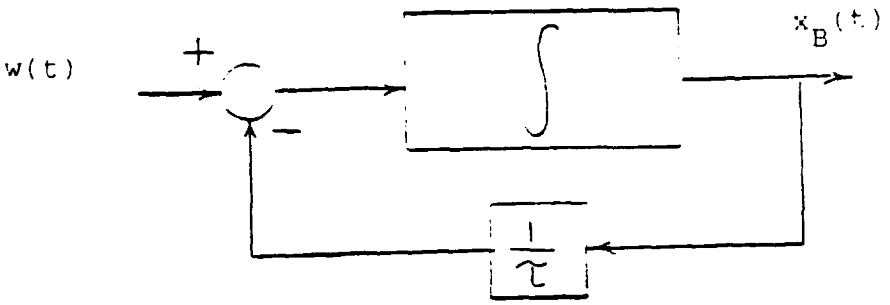
$$\underline{K}(t_i) = \underline{P}(t_i^-)\underline{H}^T(t_i)[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]^{-1} \qquad (2\text{-}22)$$

For this study we will use a one dimensional beam model. Thus, all the previous equations are reduced to the scalar case. For the beam dynamics model (F) in Equation (2-3), we will assume a first-order Gauss-Markov position process, and thus:

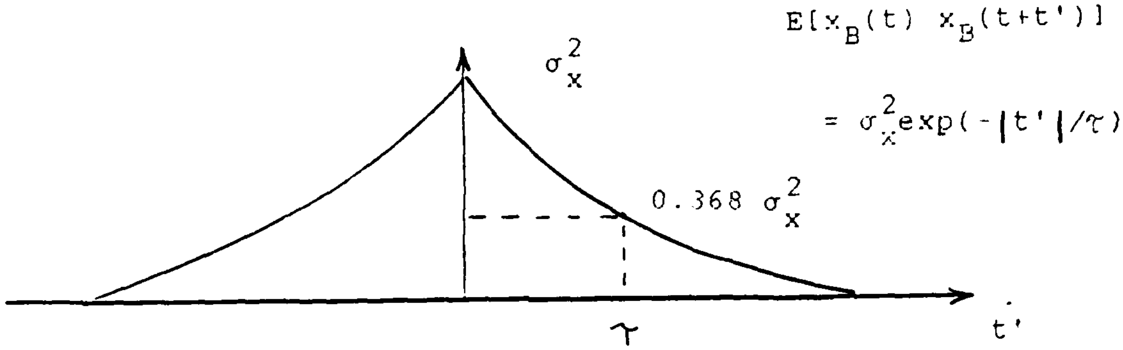
$$F = -1/\tau_B \qquad (2\text{-}23)$$

A first-order Gauss-Markov process often provides an adequate description of bandlimited processes. Figure 2.5 depicts the nature of a first-order Gauss-Markov process.

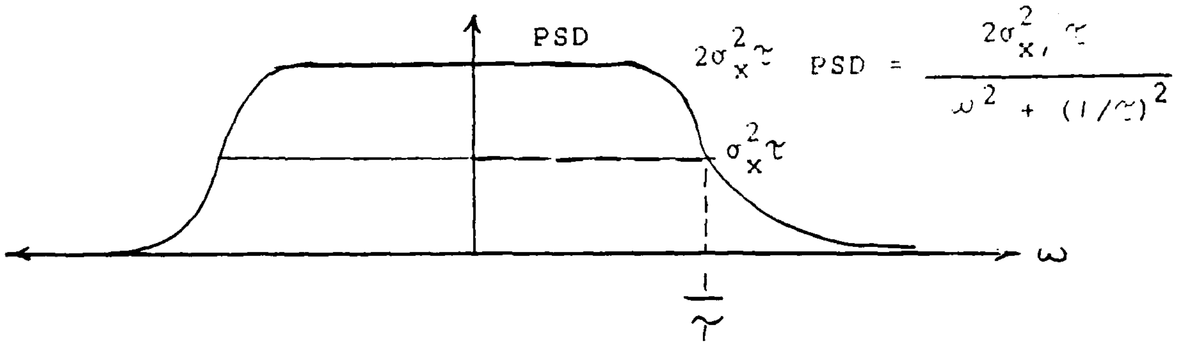
The equations for this filter look remarkably similar to the Kalman Filter equations [19,21,27,29]. The primary difference is that, the time that measurements will arrive is not known a priori. In 1978, Santiago used the Snyder-Fishman filter to investigate the performance of optical trackers when operating at low signal rates [27]. Santiago found that the Snyder-Fishman filter performance deteriorated markedly in the presence of noise. The noise



a) Beam Model



b) Autocorrelation of Output Process $x_B(t)$



c) Power Spectral Density (PSD) of $x_B(t)$
(ω = Frequency)

Figure 2.5 First Order Gauss-Markov Process

in this case was that due to dark currents within the photodetector as well as background noise. In an effort to enhance the Snyder-Fishman filter performance, Santiago [27] investigated the use of residual monitoring techniques to perform preliminary rejection of noise events. In an alternate approach, Meer [21] developed the Meer filter by applying multiple model adaptive estimation techniques to this problem. The next section will describe multiple model adaptive estimation.

2.4 Multiple Model Adaptive Estimation

As mentioned in the last section, Meer applied Multiple Model Adaptive Estimation (MMAE) techniques to solving the problem of estimating the beam position with Snyder-Fishman filters in the presence of noise, adaptively determining whether each photoelectron event is signal-induced or noise-induced. MMAE techniques are applied to the problem of estimating the beam centroid location with an uncertain beam time constant, adaptively determining the true value of the beam time constant. This section will present a development of the MMAE prior to presenting the applications of this technique in later sections.

One method of developing a MMAE of the Meer filter is to use a Bayesian approach. Following the development by Maybeck [17:129-136], the Bayesian state estimate can be expressed as the expected value of the state, conditioned on the measurement history $\underline{Z}_i$:

$$\hat{\underline{x}}(t_i^+) = E[\underline{x}(t_i)|\underline{Z}_i] = \int_{-\infty}^{\infty} \underline{\varepsilon}\, f_{x(t)|Z}(\underline{\varepsilon}|\underline{Z}_i)d\underline{\varepsilon} \qquad (2\text{-}24)$$

where $\underline{\varepsilon}$ is a dummy variable of integration of $\underline{x}(t)$, and $f_{x(t)|Z}(\underline{\varepsilon}|\underline{Z}_i)$ is the conditional probability density function of $\underline{x}(t)$, given that the realization $\underline{Z}_i$ of the measurement history $\underline{Z}$ has been observed up to the current time. If we let $\underline{a}(t)$ denote the vector of uncertain parameters (beam time constant for example), and assume $\underline{a}(t)$ can take on any value in the continuous range of A, the last equation can be rewritten to include $\underline{a}(t)$:

$$\hat{\underline{x}}(t_i^+) = \int_{-\infty}^{\infty} \underline{\varepsilon}\, [\int_A f_{x(t),a|Z}(\underline{\varepsilon},\underline{\alpha}|\underline{Z}_i)d\underline{\alpha} \;] \; d\underline{\varepsilon} \qquad (2\text{-}25)$$

where $\underline{\alpha}$ is the dummy variable used to integrate $\underline{a}(t)$ over the range of A. According to Bayes' rule, the conditional probability density function can be expressed as:

$$f_{x(t),a|Z}(\underline{\varepsilon},\underline{\alpha}|\underline{Z}_i) = f_{x(t)|a,Z}(\underline{\varepsilon}|\underline{\alpha},\underline{Z}_i)\, f_{a|Z}(\underline{\alpha}|\underline{Z}_i) \qquad (2\text{-}26)$$

The first density function on the right hand side of the last equation is the density of the state (equal to the beam centroid location since $H = 1$), given the beam time constant and measurement histories. It has a mean of $\hat{\underline{x}}(t_i^+)$ and a covariance of $\underline{P}(t_i^+)$ as computed by a Meer filter based upon a particular $\underline{a}$ or beam time constant. The second term is the conditional probability of $\underline{a}$ conditioned on the measurement history. According to Bayes' rule, it can be expressed as:

$$f_{a|Z}(\underline{\alpha}|\underline{Z}_i) = f_{a|z(i),Z(i-1)}(\underline{\alpha}|\underline{\mu}_i,\underline{Z}_{i-1})$$

$$= \frac{f_{a,z(i)|Z(i-1)}(\underline{\alpha}|\underline{\mu}_i,\underline{Z}_{i-1})}{f_{z(i)|Z(i-1)}(\underline{\mu}_i|\underline{Z}_{i-1})}$$

$$= \frac{f_{z(i)|a,Z(i-1)}(\underline{\mu}_i|\underline{\alpha},\underline{Z}_{i-1}) \quad f_{a|Z}(\underline{\alpha}|\underline{Z}_{i-1})}{\int_A f_{z(i)|a,Z(i-1)}(\underline{\mu}_i|\underline{\alpha},\underline{Z}_{i-1}) \quad f_{a|Z}(\underline{\alpha}|\underline{Z}_{i-1})d\underline{\alpha}} \tag{2-27}$$

where $f_{z(i)|a,Z(i-1)}(\underline{\mu}_i|\underline{\alpha},\underline{Z}_{i-1})$ is Gaussian, with a mean of $\underline{H}(t)\hat{\underline{x}}(t_i^-)$ and a covariance of $[\underline{H}(t_i)\underline{P}(t_i^-)\underline{H}^T(t_i) + \underline{R}(t_i)]$, again as produced by a Meer filter based on a specific value of $\underline{a}$. Here $\underline{z}(i)$ is the current measurement and $\underline{Z}_{i-1}$ is the history of all previous measurements.

The last equation could conceptually be evaluated iteratively starting from $f_a(\underline{\alpha})$. By combining equation (2-25) with equation (2-26) and switching the order of integration, we can generate the state estimate from:

$$\hat{\underline{x}}(t_i^+) = \int_A \left[\int_{-\infty}^{\infty} \underline{\xi} \; f_{x(t)|a,Z}(\underline{\xi}|\underline{\alpha},\underline{Z}_i)d\underline{\xi}\right] f_{a|Z}(\underline{\alpha}|\underline{Z}_i)d\underline{\alpha} \tag{2-28}$$

The term within the brackets of Equation (2-28) is the output from a single filter. Unfortunately, the outer integration implies an uncountably infinite number of such filters (one for every possible beam time constant). This will make this approach computationally impractical, so we let the uncertain parameter vector (again in this case the beam time constant) assume only a finite number of values, such as the discrete vector set $[\underline{a}_1, \underline{a}_2, \ldots \underline{a}_K]$, where K is the number of elemental filters in the MMAE. Thus, the integration is replaced by a finite summation. This becomes the foundation of the multiple model adaptive structure, where each estimate within the bank of estimators is based on an assumed discrete parameter value, in this case a beam time constant.

To adopt the Bayesian approach to a multiple model structure, an a priori density function is required for $\underline{a}_k(t_0)$ (to serve as an initial condition on the iteration to be discussed next):

$$f_{a(t_0)}(\underline{\alpha}) = \sum_{k=1}^{K} p_k(t_0)\, \delta(\underline{\alpha} - \underline{a}_k) \qquad (2\text{-}29)$$

where $p_k(t_0)$ is the probability that $\underline{a}$ assumes the value $\underline{a}_k$ at time t_0. Therefore, the hypothesis conditional probability, $p_k(t_i)$, is a recursive relationship expressed as:

$$p_k(t_i) = \mathrm{prob}[\underline{a}=\underline{a}_k \mid \underline{Z}(t_i)=\underline{Z}_i]$$

$$= \frac{f_{z(i)|a,Z(i-1))}(\underline{\mu}_i \mid \underline{a}_k, \underline{Z}_{i-1}) \quad p_k(t_{i-1})}{\Sigma_{j=1}^{K} \; f_{z(i)|a,Z(i-1)}(\underline{\mu}_i \mid \underline{a}_j, \underline{Z}_{i-1}) \quad p_j(t_{i-1})} \qquad (2\text{-}30)$$

where $f_{z(i)|a,Z(i-1)}(\underline{\mu}_i|\underline{a}_k,\underline{Z}_{i-1})$ can be evaluated as:

$$f_{z(i)|a,Z(i-1)}(\underline{\mu}_i|\underline{a}_k,\underline{Z}_{i-1}) = [(2\pi)^{m/2}|\underline{A}_k|^{1/2}]^{-1} \exp[-(1/2)\underline{r}_k^T(t_i)\underline{A}_k(t_i)^{-1}\underline{r}_k(t_i)] \tag{2-31}$$

where

$\underline{r}_k(t_i)$ is the residual in the k-th filter;

$$\underline{r}_k(t_i) = \underline{z}_i - \underline{H}_k(t_i)\hat{\underline{x}}_k(t_i^-) \tag{2-32}$$

$$\underline{A}_k(t_i) = \underline{H}_k(t_i)\underline{P}_k(t_i^-)\underline{H}_k^T(t_i) + \underline{R}_k(t_i) \tag{2-33}$$

m is the number of measurements at time t_i; the dimension of $\underline{r}_k(t_i)$.

Therefore, the conditional mean is the sum of the probabilistically weighted state estimates from each estimator within the bank:

$$\hat{\underline{x}}(t_i^+) = E[\underline{x}(t_i)|\underline{Z}(t_i)=\underline{Z}_i] \tag{2-34a}$$

$$\hat{\underline{x}}(t_i^+) = \sum_{k=1}^{K} \hat{\underline{x}}_k(t_i^+)\ p_k(t_i) \tag{2-34b}$$

Figure 2.6 depicts the structure of the MMAE estimator.

The preceding discussion was presented in terms of a

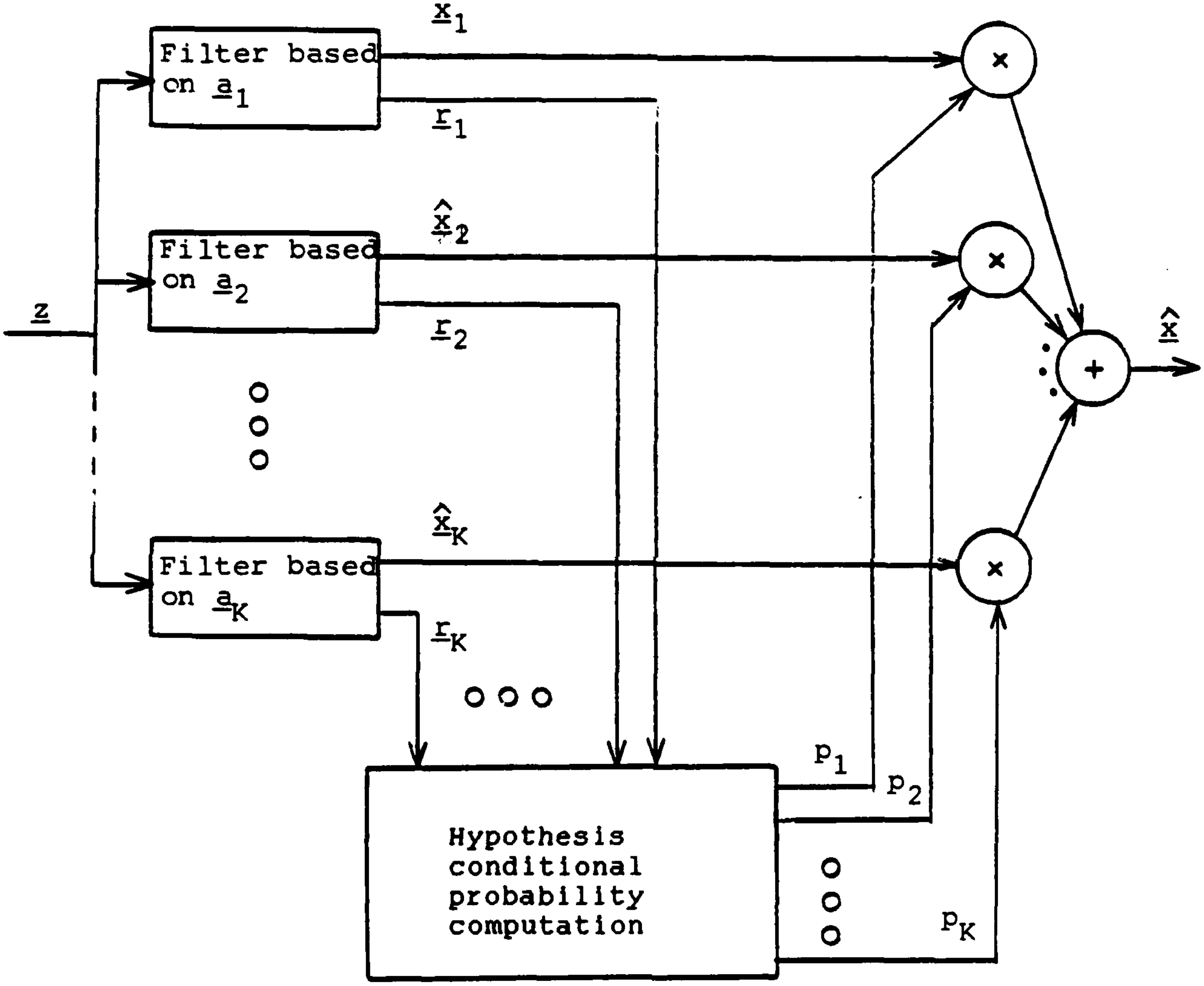


Figure 2.6 Multiple Model Filtering Algorithm

multiple model structure of elemental Meer filters to address adaptation on beam time contant τ_B. Applying this same structure, Meer developed an adaptive filter, using elemental Snyder-Fishman filters, to estimate the NPB centroid despite the presence of noise in the detector.

2.5 Meer Filter

In 1982, David Meer developed an adaptive estimator to solve the problem of estimating the NPB in the presence of noise [21]. This section will begin by presenting the basic Meer filter. Then, the methods applied by Meer, and later Zicker, to reduce the computational load of a simple Meer filter, will be presented.

2.5.1 Basic Meer Filter. The basic Meer filter is an application of the MMAE structure with elemental Snyder-Fishman filters. Each model (Snyder-Fishman filter) in this structure is based on a separate hypothesis sequence for the order of incoming signal and noise events. The order of the occurrence of signal and noise events specifies each individual model's hypothesis sequence as shown in Table I. For each event labeled as a signal, the information (location of the event on the detector array at that time) is used to update the beam location estimate of the filter based on that hypothesis sequence. This update is performed using Equations (2-20) and (2-21). For each event labeled as a noise, no update occurs so that:

Table 2.1 Hypothesis Sequences

Number of Events	0	1	2	3
Possible Sequences				SSS
				SSN
				SNS
				SNN
			SS	NSS
			SN	NSN
	–	S	NS	NNS
		N	NN	NNN
Total Number of Hypothesis Sequences	0	2	4	8

$$\hat{\underline{x}}_j(t^+) = \hat{\underline{x}}_j(t^-) \quad (2\text{-}35)$$

$$\underline{P}_j(t^+) = \underline{P}_j(t^-) \quad (2\text{-}36)$$

where $\hat{\underline{x}}_j$ is the conditionally expected value of the beam states for the j-th hypothesis and $\underline{P}_j$ is the variance for the j-th hypothesis sequence. The state estimate of each elemental filter is expressed as:

$$\hat{\underline{x}}_j(t) = E[\underline{x}(t) | h_j^{N_t}, Z^{N_t}] \quad (2\text{-}37)$$

where $\hat{\underline{x}}_j(t)$ is the expected value of the beam states, $\underline{x}(t)$, conditioned on the j-th assumed hypothesis time history $h_j^{N_t}$ and the observations history, Z^{N_t}, of events $(t_i, \underline{z}_i)$, where $i = 1,2,\ldots,N_t$.

The sequences or hypothesis time histories can be formed into conditional probability trees to illustrate Bayes' rule (See Figure 2.7). This study will use the convention that even numbered hypothesis sequences have assumed the most recent event, or N_t-th event, was a signal. For a hypothesis sequence corresponding to the assumption that the most recent event is signal-induced, the conditional probability is given by $(j = 2,4,\ldots 2^{N_t})$:

$$Pr(h_{2j} | Z^{N_t}) = Pr(s)\, Pr(h_j | Z^{N_t - 1}) \quad (2\text{-}38)$$

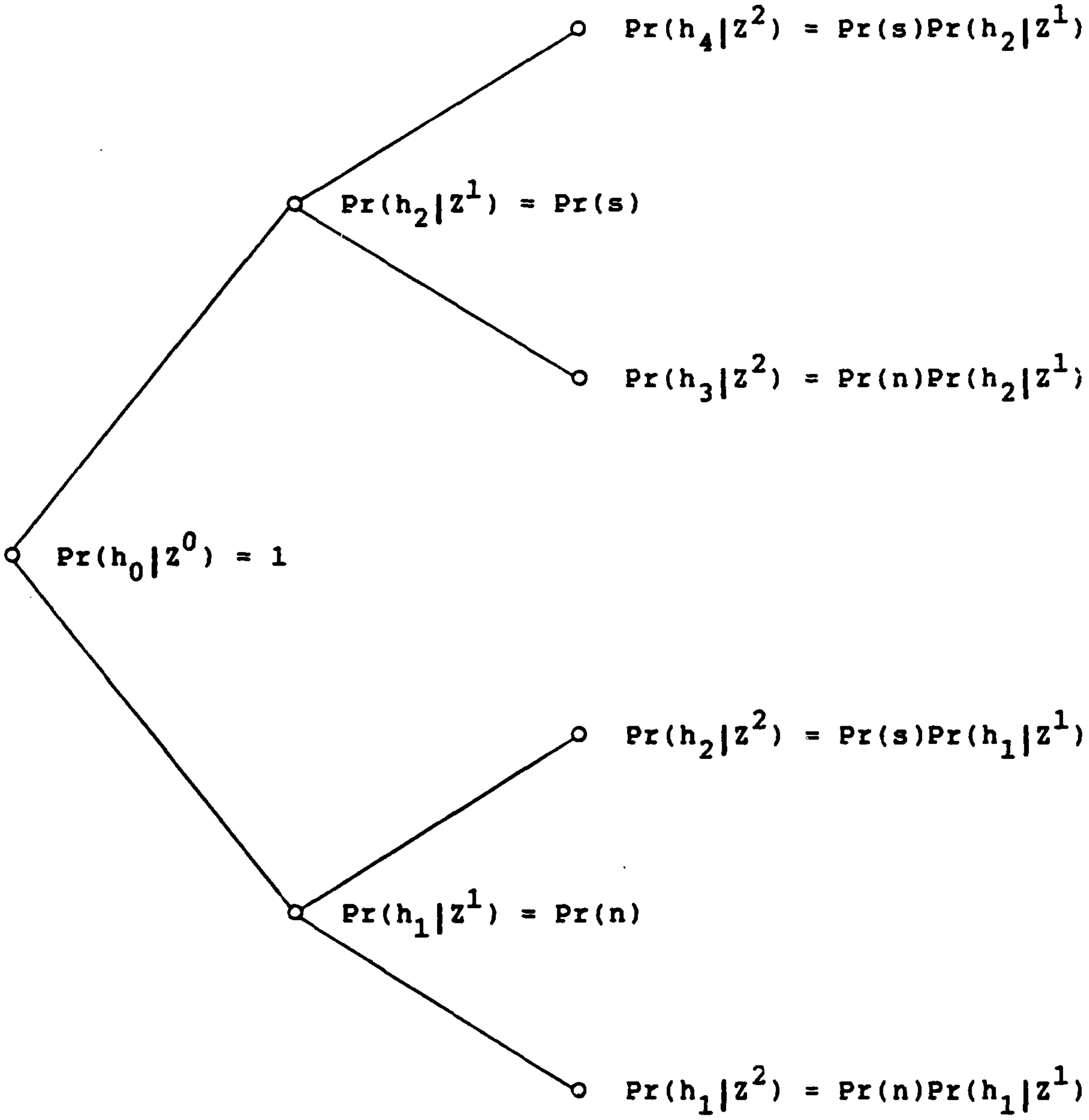

$Pr(h_4|Z^2) = Pr(s)Pr(h_2|Z^1)$
$Pr(h_2|Z^1) = Pr(s)$
$Pr(h_3|Z^2) = Pr(n)Pr(h_2|Z^1)$
$Pr(h_0|Z^0) = 1$
$Pr(h_2|Z^2) = Pr(s)Pr(h_1|Z^1)$
$Pr(h_1|Z^1) = Pr(n)$
$Pr(h_1|Z^2) = Pr(n)Pr(h_1|Z^1)$

Figure 2.7 Conditional Probability Tree

where there are now 2j signal related hypothesis sequences as shown in Figure 2.7 and

$$\Pr(s) = \frac{\lambda_s(t_{N_t}, r_{N_t}, \hat{x}(t_{N_t}))}{\lambda_s(t_{N_t}, r_{N_t}, \hat{x}(t_{N_t})) + \lambda_n(t_{N_t}, r_{N_t})} \qquad (2\text{-}39)$$

where λ_s and λ_n are the signal and noise rate parameters previously described in Section 2.2. The weighting probability starts at t_0, with the initial conditional probability defined as $\Pr(h_0 | Z^0) = 1$.

For hypothesis sequences assuming the N_t-th event was noise induced, the conditional probability is given by ($j = 1,3,\ldots 2^{N_t}-1$):

$$\Pr(h_{2j-1} | Z^{N_t}) = \Pr(n) \Pr(h_j | Z^{N_t-1}) \qquad (2\text{-}40)$$

where

$$\Pr(n) = \frac{\lambda_n(t_{N_t}, r_{N_t})}{\lambda_s(t_{N_t}, r_{N_t}, \hat{x}(t_{N_t})) + \lambda_n(t_{N_t}, r_{N_t})} \qquad (2\text{-}41)$$

The overall state estimate of the MMAE is the probabilistically weighted average of individual filter state estimates given by:

$$\hat{\underline{x}}(t) = \sum_{j=1}^{2^{N_t}} \Pr(h_j^{N_t} | Z^{N_t}) \, \hat{\underline{x}}_j(t) \qquad (2\text{-}42)$$

with an MMAE computed error variance defined as:

$$\underline{P}(t) = \sum_{j=1}^{2^{N_t}} \Pr(h_j^{N_t} | Z^{N_t}) \, (\underline{P}_j(t) + [\hat{\underline{x}}_j(t) - \hat{\underline{x}}(t)] \, [\hat{\underline{x}}_j(t) - \hat{\underline{x}}(t)]^T) \qquad (2\text{-}43)$$

The structure of the Meer Filter is depicted in Figure 2.8.

Note that the number of hypothesis sequences, and therefore the number of models or Snyder-Fishman filters, required to estimate the beam location optimally by the occurrence of N_t events is 2^{N_t}. This quickly becomes a computationally burdensome, and in fact intractable, filter. For example, ten events requires $2^{10} = 1024$ models to implement the basic filter fully. Meer recognized this drawback of the basic filter design [21] and proceeded to develop a method to limit the expansion of the hypothesis tree by pruning some of the hypothesis sequences.

2.5.2 Pruning the Hypothesis Tree. A way to limit the growth of, or prune, the hypothesis tree was needed. Meer proposed a number of possible methods to prune the hypothesis tree and chose the "Best Half Method" as the most viable [21]. Zicker [32], applied the work of Weiss, Upadhyay, and Tenney [30,31] to limit the expansion of the hypothesis tree by combining the branches using the "Merge Method".

2.5.2.1 Best Half Method. Meer limited the

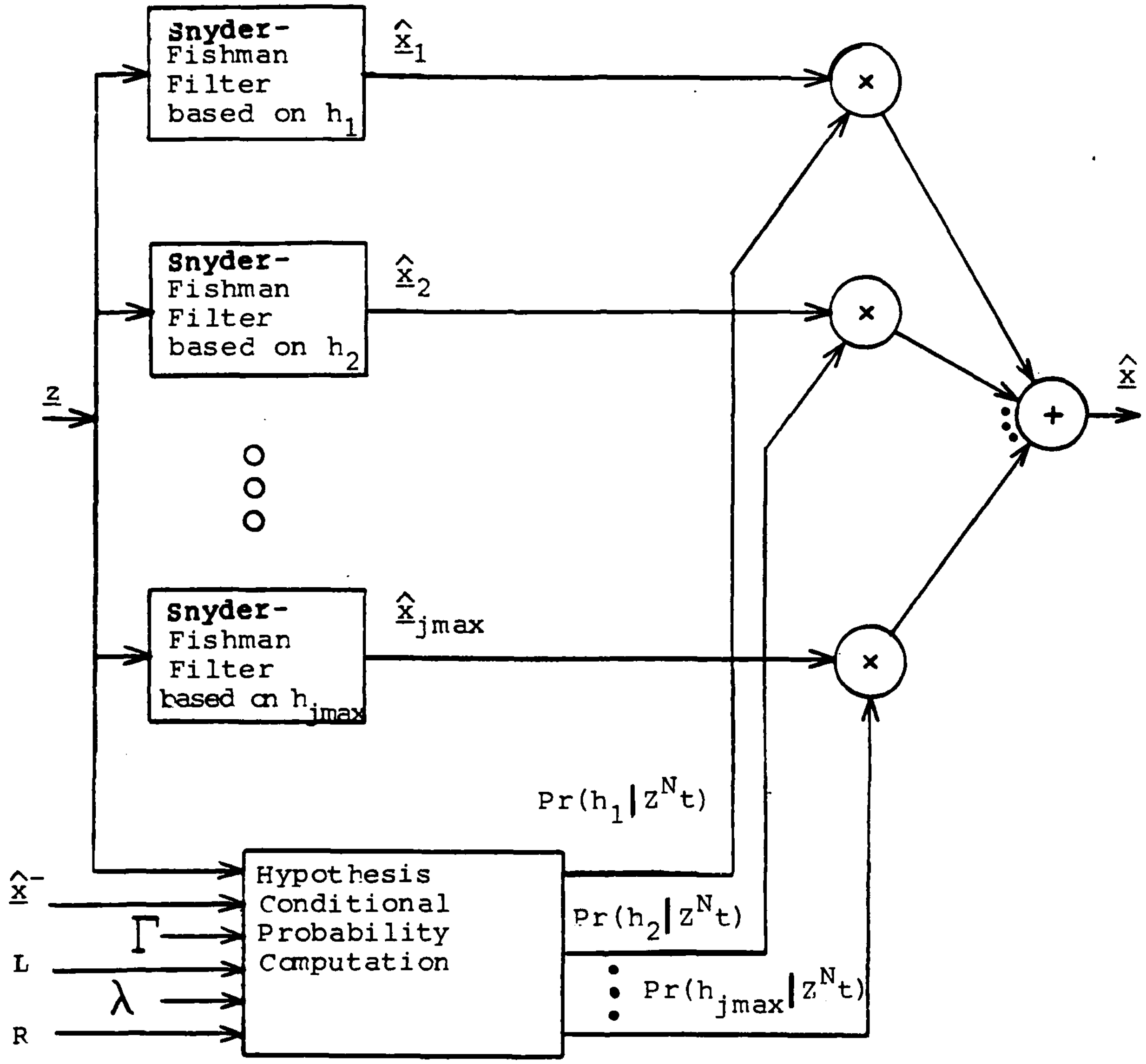

Snyder-Fishman Filter based on h_1
Snyder-Fishman Filter based on h_2
Snyder-Fishman Filter based on h_{jmax}
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_{jmax}$
z
×
+
$\hat{x}$
$Pr(h_1|Z^{N}t)$
$Pr(h_2|Z^{N}t)$
$Pr(h_{jmax}|Z^{N}t)$
Hypothesis Conditional Probability Computation
$\hat{x}^-$
Γ
L
λ
R

Figure 2.8 Meer Filter Structure (Note: $jmax=2^{N}t$)

expansion of the hypothesis tree by limiting the number of events, or the history of events, that each hypothesis sequence is allowed to consider. This number of events defines the memory depth D of the suboptimal Meer filter. For this study, the following convention is used to describe the age of an event.

> When an event A first occurs it has an age of 1. When the next event occurs Event A's age becomes 2. Event A will continue to age, as new events come in, until its age equals the depth. When the age of Event A reachs depth, it is the oldest event. When the next event occurs, Event A is discarded.

Once the Meer filter reachs the prescribed depth, a comparison is made between the probability of the oldest event being a signal and the probability the oldest event being a noise. Conceptually, if the comparison indicates the oldest event was most probably a signal, then the sequences associated with the oldest event being a signal (the upper half of the tree) are retained (See Figure 2.9a). If the comparison indicates the oldest event was most probably a noise, then the sequences associated with the oldest event being a noise (the lower half of the tree) are retained (See Figure 2.9b). The oldest event and its probabilities are then replaced by the event and probabilities of the next-oldest event. The process of selecting the best half would then be repeated as each event comes in. Thus, only hypothesis sequences associated with the most recent "depth D" number of events are retained.

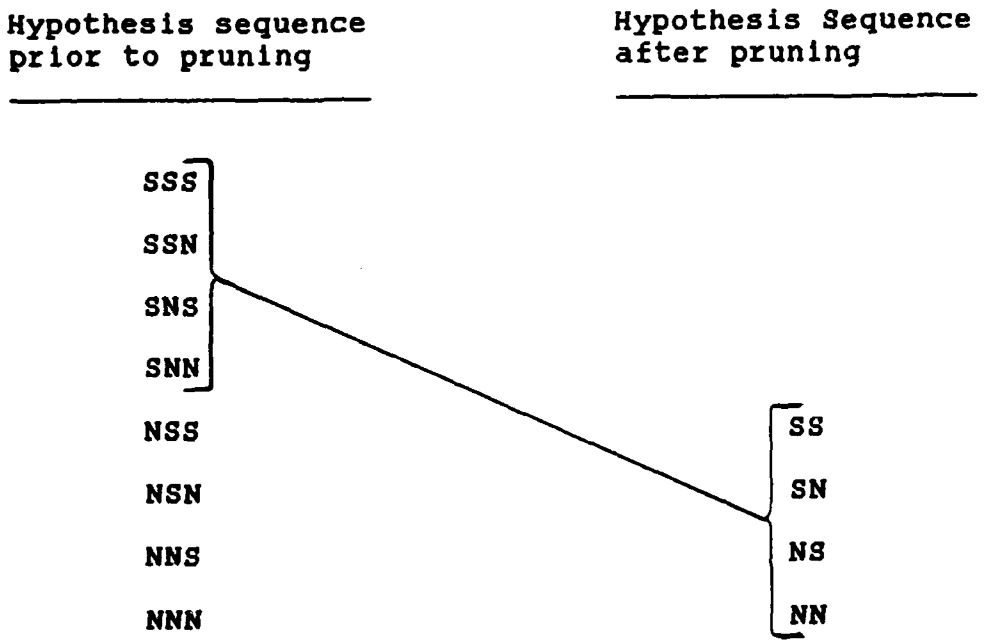

Hypothesis sequence prior to pruning
Hypothesis Sequence after pruning
SSS
SSN
SNS
SNN
NSS
NSN
NNS
NNN
SS
SN
NS
NN

Figure 2.9a Best Half Pruning with Signal Most Probable. (Depth = 3)

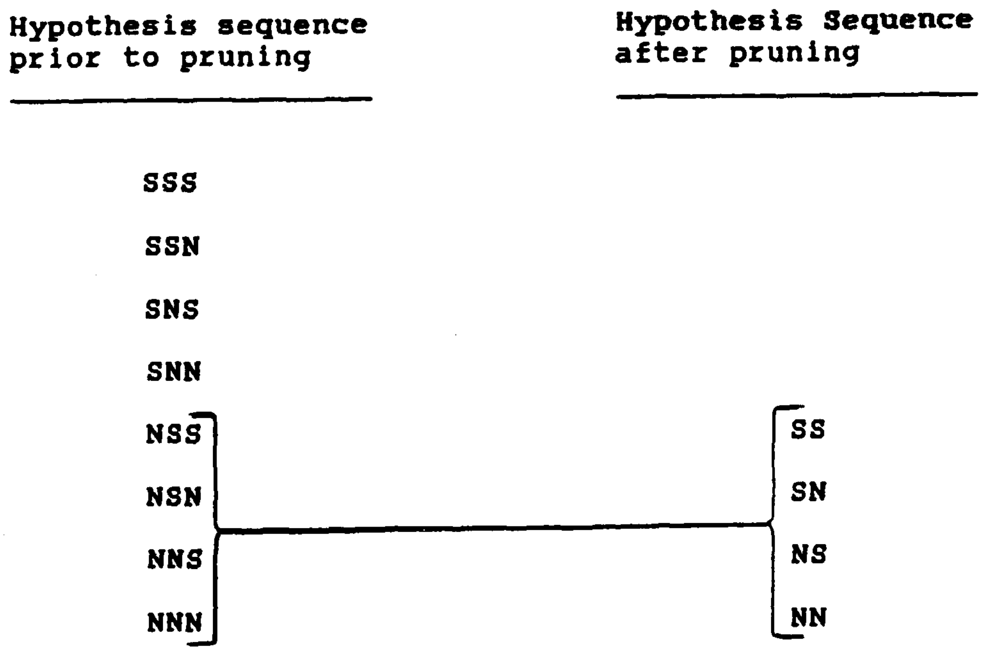

Hypothesis sequence prior to pruning
Hypothesis Sequence after pruning
SSS
SSN
SNS
SNN
NSS
NSN
NNS
NNN
SS
SN
NS
NN

Figure 2.9b Best Half Pruning with Noise Most Probable. (Depth = 3)

Although the complete hypothesis sequences are lost, the portion of the sequence that is discarded still has an influence on the hypothesis time history (i.e. which set of hypothesis sequences are retained). Conceptually, although an event only directly impacts the estimation of the beam state for "depth D" events (while it is in the filter), it indirectly impacts state estimation after leaving the filter by determining which sequence will be retained for future beam state estimation.

2.5.2.2 The Merge Method. In 1983, Zicker incorporated the work of Weiss, Upadhyay, and Tenney [30,31] to form a different method of limiting the hypothesis expansion. The purpose of this method is to retain a larger portion of the entire measurement history [32:26]. Like the "Best Half Method", the number of sequences is limited to a prescribed depth. In this case, however, the retained estimates are determined from the probabilistically weighted sum of the estimates from hypothesis sequences that differ only in the assumption that the oldest event in the filter was a signal or noise (See Figures 2.10 and 2.11). The probabilistic weight used is the probability associated with the oldest event in the filter being a signal or noise. Similarly, the retained variances are determined using the probablistic weighting of the oldest event. In equation form this becomes:

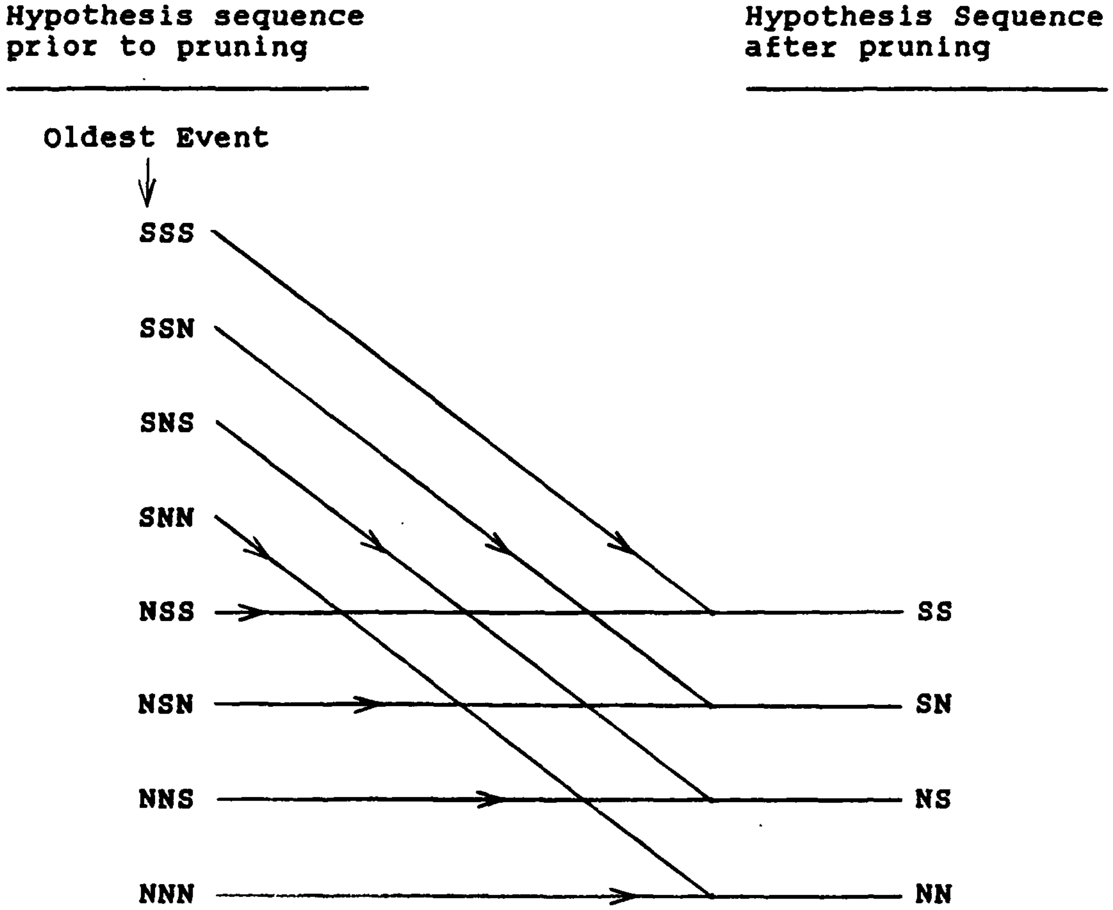


Figure 2.10 Merge Method Hypothesis Merging (Depth = 3)

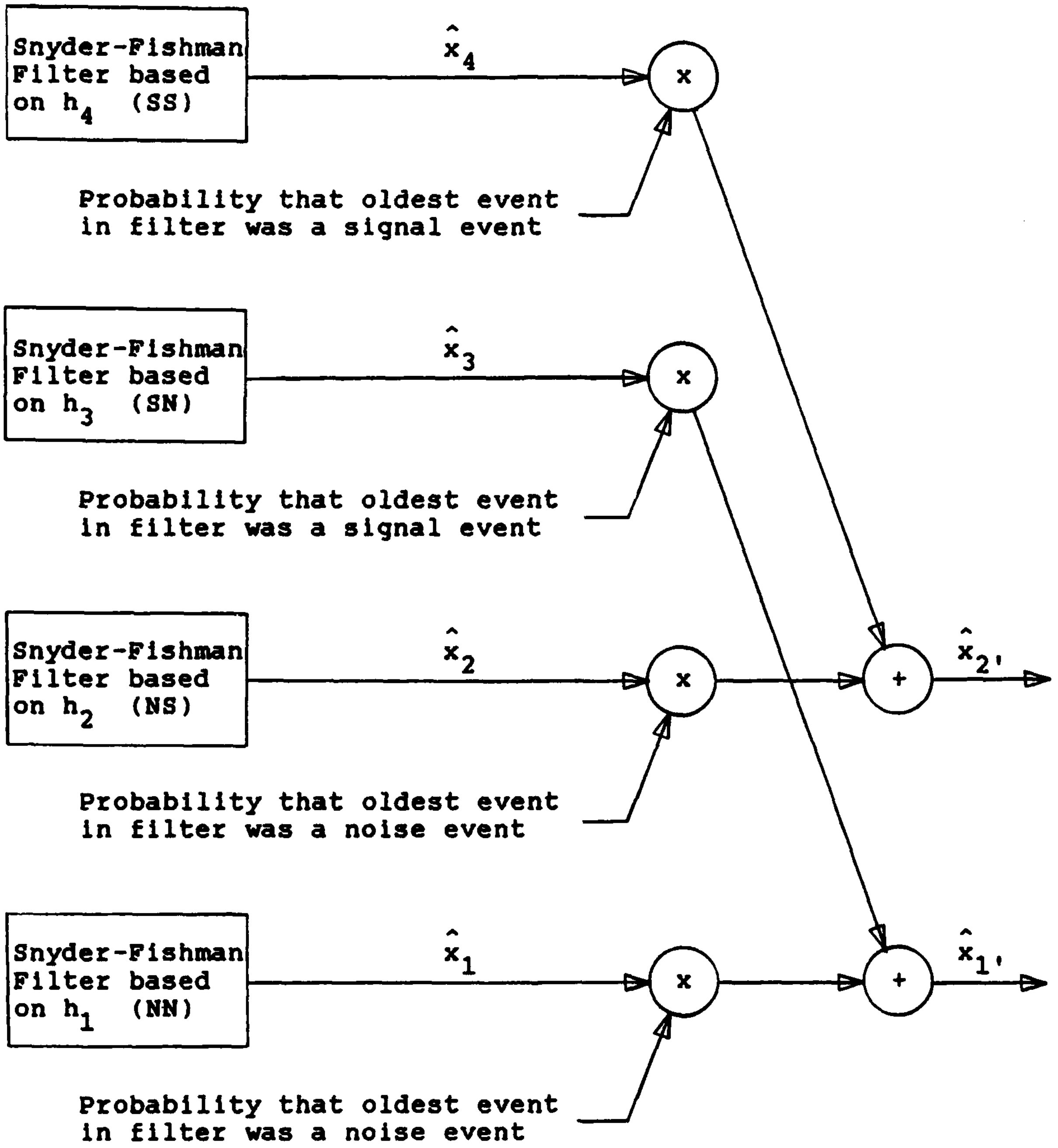

Snyder-Fishman Filter based on h_4 (SS)
$\hat{x}_4$
x
Probability that oldest event in filter was a signal event
Snyder-Fishman Filter based on h_3 (SN)
$\hat{x}_3$
x
Probability that oldest event in filter was a signal event
Snyder-Fishman Filter based on h_2 (NS)
$\hat{x}_2$
x
+
$\hat{x}_{2'}$
Probability that oldest event in filter was a noise event
Snyder-Fishman Filter based on h_1 (NN)
$\hat{x}_1$
x
+
$\hat{x}_{1'}$
Probability that oldest event in filter was a noise event

Figure 2.11 Merge Method Pruning (Depth = 2)

$$\underline{x}_{j'}(t) = Pr(oldest=n)\underline{x}_j(t)+Pr(oldest=s)\underline{x}_k(t) \qquad (2\text{-}44)$$

$$\begin{aligned}\underline{P}_{j'}(t) = [Pr(oldest=n)\ (\underline{P}_j(t) + [\underline{x}_j(t) - \underline{x}_{j'}(t)]\\ [\hat{\underline{x}}_j(t) - \hat{\underline{x}}_{j'}(t)]^T) + Pr(oldest=s)(\underline{P}_k(t)\\ + [\hat{\underline{x}}_k(t) - \hat{\underline{x}}_{j'}(t)][\hat{\underline{x}}_k(t) - \hat{\underline{x}}_{j'}(t)]^T) \qquad (2\text{-}45)\end{aligned}$$

Note that Pr(oldest=n) + Pr(oldest=s) = 1

where

$\underline{x}_j$ and $\underline{x}_k$ are the estimates associated with hypothesis that differ only in the decision that the oldest event in the filter is a noise or signal, respectively. Here, j is associated with the sequence that assumed a noise event occurred and k is associated with the sequence that assumed a signal event occurred. Thus, $k = j + 2^{D-1}$ and $j = 0,1,...,(2^{D-1})$.

D is the memory depth of the hypothesis tree before the pruning process has occurred.

$\underline{x}_{j'}$ and $\underline{P}_{j'}$ are the "merged" state estimates and error covariances as the number of elemental filters are reduced from 2^D to 2^{D-1}; $j' = 0,1,...,(2^{D-1})$. These are used for propagation to the next measurement time, hence the prime (') designation.

In this manner the information contained the both halves of the hypothesis tree are retained to a greater degree than in the best half method.

One unique aspect of the Merge Method as first noted by Zicker [32] is the simplification possible when the Meer filter depth is reduced to one. The Meer filter probabilistic weighting Equation (2-42) can be combined with the Snyder-Fishman update Equation (2-20) to form a single Meer filter update equation:

$$\hat{x}(t_i^+) = \hat{x}(t_i^-) + Pr(s)K(t_i)[r_i - H(t_i)\hat{x}(t_i^-)] \quad (2\text{-}46)$$

The Meer filter was designed to acheive the objective of estimating the NPB centroid in the presence of noise. The next section will present a MMAE using a bank of these Meer filters to allow the estimation of the NPB centroid with an uncertain beam time constant.

2.6 MMAE with Meer Elemental Filters

As Meer pointed out [21:170], it is important to have an accurate propagation model as the estimate of $\hat{x}(t_i^-)$ is used to approximate λ_s by Equation (2-7) with x(t) replaced by $\hat{x}(t)$. If the assumed model of the underlying process has characteristics different from those of the actual physical process, then the approximation of λ_s may be inadequate. Since the beam time constant τ_B is used to propagate the estimate of x via Equations (2-15) and (2-16) with $F = -1/\tau_B$, an uncertain beam time constant can result in an inadequate approximation of λ_s. Under these conditions we we would have degradation of performance. In other words, the filter may have poor robustness. For a filter variation of only 2% from the true beam time constant, Johnson found significant degradation in performance [9]. Johnson recommended MMAE techniques be used for online adaptation of the beam time constant. The concept here is to take a set of Meer filters, each modeled on a different beam time constant, combine the outputs with weighting based on residual performance to arrive at a better estimate of the state and an estimate of the actual beam time constant. The MMAE structure presented in Section 2.4 supports the weighted estimation of the state. The same structure also supports the weighted estimation of the beam time constant. If we can estimate the correct beam time constant, then we can propagate the filter state estimate with greater accuracy by using the beam time constant estimate in our model. The better state estimate provides a better estimate of λ_s through Equation (2-7). This, in turn, will contribute to better state and beam time constant estimates. A well propagated state estimate is also important as an input to a state feedback beam controller. Similarly, the estimate of the beam time constant could be used in an adaptive controller to determine the appropriate controller gains. Thus, it is desirable to seek an estimate of the beam time constant. The conditional mean of the beam time constant is given by [17:132-133]:

$$\hat{\tau}(t_i) = E\{\tau(t_i) | Z(t_i)=Z_i\}$$

$$= \Sigma_{k=1}^{K} \tau_k \; p_k(t_i) \qquad (2\text{-}47)$$

where τ_k is the individual beam time constant associated with each Meer filter, and p_k is found through Equation (2-30). An indication of the precision of this estimate is given by the conditional variance [17:133]:

$$P_\tau = E\{(\tau_k - \hat{\tau}(t_i))^2 | Z(t_i)=Z_i \}$$

$$= \Sigma_{k=1}^{K} (\tau_k - \hat{\tau}(t_i))^2 \; p_k(t_i) \qquad (2\text{-}48)$$

The structure of the MMAE using elemental Meer filters to estimate the beam state is depicted in Figure 2.12. The same structure used to estimate the beam time constant is depicted in Figure 2.13.

2.7 Summary

This chapter began with a description of the NPB dynamics model. Following this, the Poisson space-time point process description was applied to the case of measurements extracted from a NPB being irradiated with laser beams. Next, the Snyder-Fishman filter was described which uses the Poisson point process as its measurement model. After this, the MMAE structure was described to

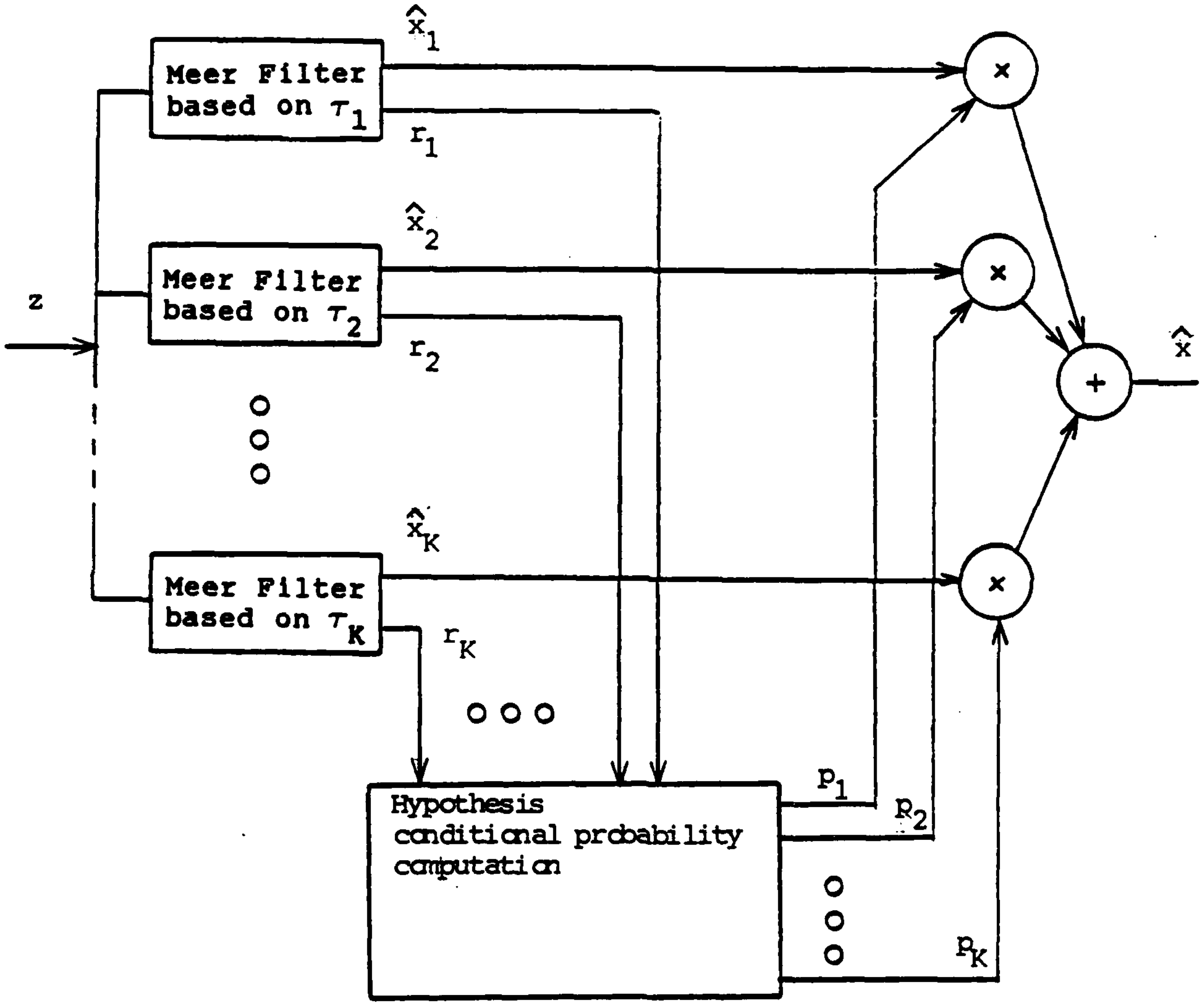

$\hat{x}_1$
Meer Filter based on τ_1
r_1
$\hat{x}_2$
z
Meer Filter based on τ_2
r_2
$\hat{x}_K$
Meer Filter based on τ_K
r_K
Hypothesis conditional probability computation
p_1
p_2
p_K
×
×
×
+
$\hat{x}$

Figure 2.12 MMAE Meer State Estimation

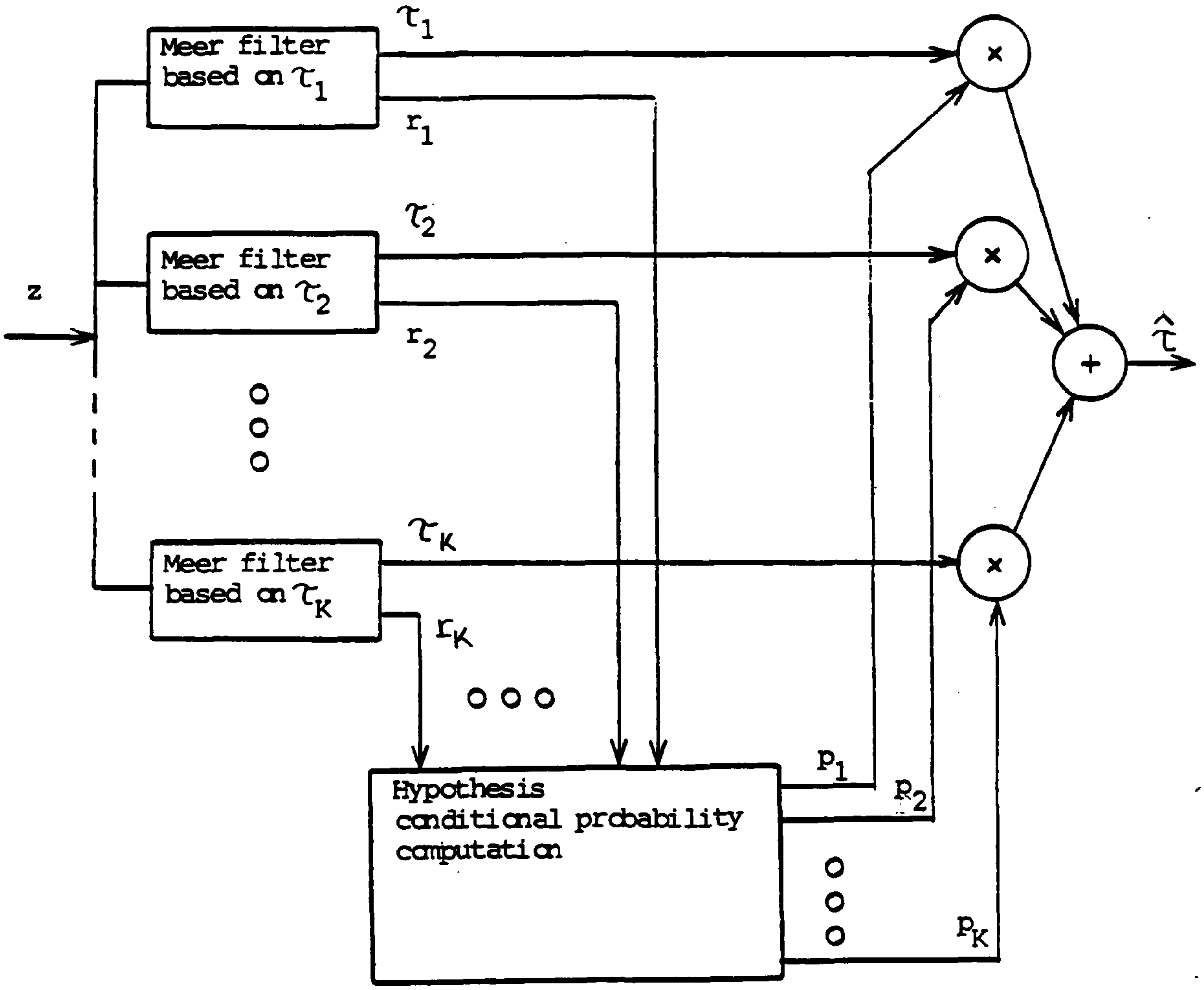

Meer filter based on τ_1
Meer filter based on τ_2
Meer filter based on τ_K
z
τ_1
r_1
τ_2
r_2
τ_K
r_K
Hypothesis conditional probability computation
p_1
p_2
p_K
×
×
×
+
$\hat{\tau}$

Figure 2.13 MMAE Meer parameter estimation

support the concepts of the Meer filter and MMAE Meer filter. The Meer filter was described as an MMAE structure which uses a bank of Snyder-Fishman filters to estimate NPB centerline in the presence of noise. Each Snyder-Fishman filter in the bank operated under a separate hypothesis for the order of signal and noise events. Tree pruning was then described as a means to reduce the computational loading caused by an ever-expanding hypothesis tree. Finally, a Multiple Model Adaptive Estimator (MMAE) using elemental Meer filters was developed as an approach to accommodate the effects of an uncertain beam time constant.

III. Controller Design

The MMAE Meer filter provides the estimates necessary to track the location of the NPB. The next step is the pointing of the NPB at a desired target. This chapter develops a controller to accomplish that purpose. The objective of the controller is to point the beam at the target. Therefore, we begin by describing the target model. Next, we will describe the controller gain synthesis technique. In each of the previous controller designs, assumed certainty equivalence methods were employed, and the controller gains were determined based on the assumptions of a linear system, deterministic optimal control law, full-state feedback, and a quadratic cost criterion. [8:Sec 3,1-2; 9:30; 23:22; 32:29]. The controller gains were determined by LQ synthesis techniques assuming beam and target states were perfectly known. Then, by assumed certainty equivalence, the full-state feedback was replaced by the beam state estimate provided by the Meer filter and the target state estimates provide by a Kalman filter. Figure 3.1 is a simple block diagram of this controller structure. This study will use a similar design process, but will also use Implicit Model Following techniques to aid the determination of appropriate controller gains. Finally, we will specify a structure for the controller. In previous thesis efforts, the following controllers were developed:

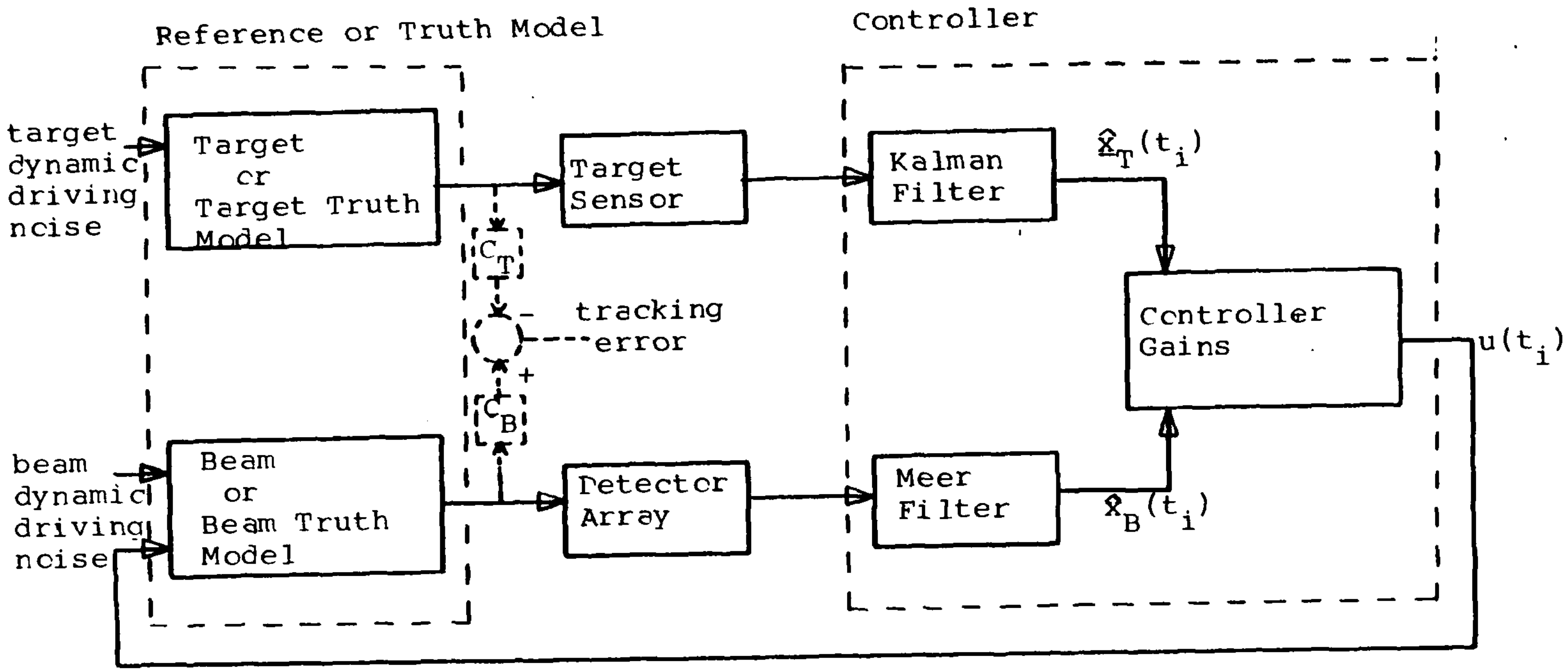


Figure 3.1 Controller Structure

the proportional gain controller, the proportional plus integral (PI) controller, and the multiple model adaptive controller with elemental controllers based on separate assumptions of the target dynamics driving noise strength. Once the controller structure is set, the synthesis techniques are used to determine the appropriate gains to be used in the selected controller.

3.1 Target Models and Filter

We include a target dynamics model in our target state estimator design to allow finer tracking of the target than possible with target measurements alone. This assumes, of course, that the actual target is well described by the target dynamics model. Even when the target is well described by our dynamics model, there are uncertainities in the dynamics driving forces and sensor measurement data that we need to consider in estimating the beam and target locations. To combine the information we have, or assume, about the target dynamics and the actual sensor measurements, we will use a Kalman filter. In incorporating a target model into the filter, we must consider the model's impact on the stability of the filter (and eventually on the stability of the closed loop system incorporating that filter). First, this section describes the target dynamics model used in this study. Then the target filter used to estimate the target states is presented.

3.1.1 Target Truth Model. Our objective is to determine whether a controller, using beam state estimates provided by a Meer filter, can point the NPB at a desired target. Therefore we need a target model. In this section, the target truth model used in this study is presented. As this is yet a feasibility study, we have a wide latitude in the choice of a target model. Our desire is a target which is realistic and stresses the NPB controller performance. A realistic model would include position, velocity, acceleration as functions of time, as these are commonly used to describe motion. As most targets travel much more slowly than the speed of light, we will ignore relativistic effects in our target model. A realistic model would also be three dimensional, having position, velocity and acceleration components in three spatial directions. Included in the model would be the dynamics driving forces in each of these directions. Since we have reduced the beam model for this study to a single dimension in space, the output of a three dimensional target model would have to be projected onto this one dimensional space. Instead, we will imploy target model with a single spatial dimension to match the reduction of the beam model to a single spatial dimension. To account for the order reduction we might incorporate additional dynamics driving noises on the motion states of the target differential equation model. Beginning with Jamerson in 1985 [8:Sec 3,11], this continuing study has incorporated a one dimensional target model having components of position, velocity and acceleration with dynamics driving noise on the acceleration state. To maintain continuity with efforts since that time, this study will use the same model without alteration.

Johnson describes this target model as a target dynamics position process that is modelled as a double integral of exponentially time correlated acceleration; a third-order Markov position process that is the output of a third order, linear, time invariant shaping filter driven by a stationary white Gaussian noise process that has a mean of zero and strength Q [9:13]. Figure 3.2 is a block diagram of the target model. The set of first-order coupled differential equations relating the first-order Gauss-Markov acceleration to the position are:

$$dx_{TP} / dt = x_{TV} \tag{3-1}$$

$$dx_{TV} / dt = x_{TA} \tag{3-2}$$

$$dx_{TA} / dt = -1/\tau_T \, x_{TA} + w_T \tag{3-3}$$

or in matrix form:

$$\begin{bmatrix} \dot{x}_{TP} \\ \dot{x}_{TV} \\ \dot{x}_{TA} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau_T \end{bmatrix} \begin{bmatrix} x_{TP} \\ x_{TV} \\ x_{TA} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_T \tag{3-4a}$$

or

$$\underline{\dot{x}}_T(t) = \underline{F}_T \underline{x}_T(t) + \underline{G}_T w(t) \tag{3-4b}$$

where

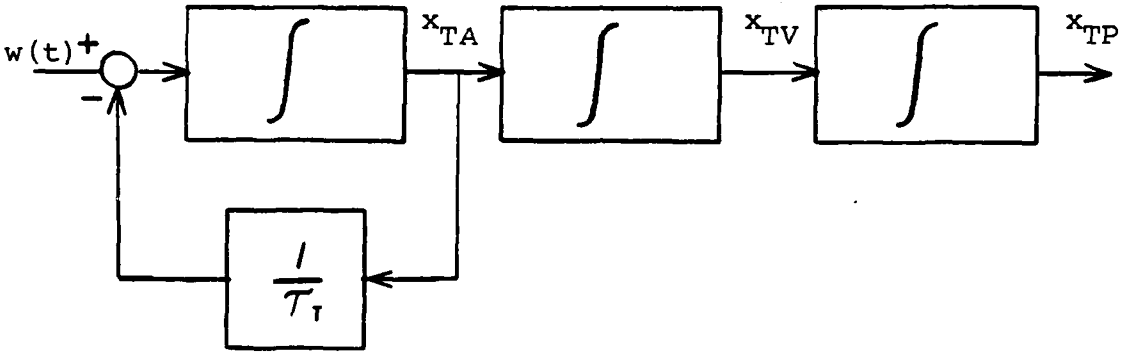


Figure 3.2 Target Truth Model

x_{TP} is the target position state

x_{TV} is the target velocity state

x_{TA} is the target acceleration state

τ_T is the acceleration correlation time constant for the target

$w_T(t)$ is a zero mean white Gaussian noise of strength Q_T, i.e. $E\left\{w_T(t)w_T(t+\alpha)\right\} = Q_T\ \delta(\alpha)$

T subscript indicates these terms are related to the target

Jamerson describes the position process generated by this model as being more difficult to track than a first-order Gauss-Markov position process, but more realistic [8:Sec 3,11] for airborne targets. Thus, subsequent thesis efforts have been based on this target model.

3.1.2 Target Filter. Since one of our objectives is to determine the effects of the Meer filter estimates on the controller, we could measure controller performance with target truth data and Meer filter beam estimates provided to the controller. This would be compared to controller benchmark performance generated by providing the controller with both beam and target truth data. This would isolate any degraded controller characteristics due to using the Meer filter estimates associated with the beam. We could also provide the controller with beam truth data and then estimates out of a target Kalman filter. This would isolate any controller characteristics due to using the target filter, but development of an optimal target filter is not the thrust of this study. However, there may also be degraded controller characteristics that result from providing both beam and target estimates to the controller that do not occur with beam estimate and target truth data. So that we can determine controller characteristics when Meer filter estimates are combined with target filter estimates, i.e., so that we can evaluate the performance potential of the eventual implementable algorithm, we must design a target filter. This section presents the target filter that provides the target estimates to the controller.

In designing our target filter we would like to exploit all the information we have concerning the target, our assumptions about the driving dynamics, and our assumptions about measurement errors committed by our sensor system. To combine this information optimally, we will use a Kalman filter. As the focus of this study is on the Meer filter, it is reasonable that the target filter model should be based on our truth model. The forms of the dynamics model and the measurement model for the target are linear time-invariant stochastic equations driven by white Gaussian noise, so the standard Kalman filter equations can be used to determine the target state estimates. Therefore, the Kalman filter state estimate and error covariance matrix propagation equations become:

$$\hat{\underline{x}}_T(t_{i+1}^-) = \underline{\Phi}_T(t_{i+1},t_i)\hat{\underline{x}}_T(t_i^+) \tag{3-5}$$

$$\underline{P}_T(t_{i+1}^-) = \underline{\Phi}_T(t_{i+1},t_i)\underline{P}_T(t_i^+)\underline{\Phi}_T(t_{i+1},t_i)$$

$$+\int_{t_i}^{t_{i+1}} \underline{\Phi}_T(t_{i+1},\alpha)\underline{G}(\alpha)Q(\alpha)\underline{G}^T(\alpha)\underline{\Phi}_T^T(t_{i+1},\alpha)\, d\alpha \tag{3-6}$$

where $\underline{\Phi}_T$ is the state transition matrix associated with $\underline{F}_T$:

$$\underline{\Phi}_T = \begin{bmatrix} 0 & \Delta t & \tau_T^2[\Delta t/\tau_T - 1 + \exp(-\Delta t/\tau_T)] \\ 0 & 1 & \tau_T[1-\exp(-\Delta t/\tau_T)] \\ 0 & 0 & \exp(-\Delta t/\tau_T) \end{bmatrix} \tag{3-7}$$

where Δt is the sample period.

We have further assumed that measurements of the target position can be made available at regular intervals rather than the random intervals as with the Meer filter. For this study, a sensor such as a radar or electro-optical sensor is assumed to provide a direct measurement of the target position states, with some measurement corruption noise added to account for nonlinear effects and variations in the index of refraction in the atmosphere (for near earth targets), sensor vibration, background radiation, non-signal-induced currents in the sensor, and fundamental limits due to the operating wavelength of the sensor. The

Central Limit Theorem supports the fact that the combined effects of these noise sources may be modelled as Gaussian. Therefore, we will model the noise as a zero mean white Gaussian noise with variance R. For simplicity, it is also assumed that these measurements can be expressed conveniently in the same coordinate frame as the beam location. To account for this explicitly would merely involve a transformation matrix. This results in a measurement equation of the form:

$$z(t_i) = \underline{H}_T(t_i)\ \underline{x}_T(t_i) + v(t_i) \qquad (3\text{-}8a)$$

or in this case,

$$z(t_i) = [1\ 0\ 0\ ]\ \underline{x}_T(t_i) + v(t_i) \qquad (3\text{-}8b)$$

where z is a scalar measurement and v is assumed to be a zero-mean white Gaussian discrete time noise with variance R. The measurement noise v is assumed to be independent of the dynamics driving noise w of the target. $\underline{H}$ reflects our assumption that the sensor measures the target position state directly. Figure 3.3 depicts the target and sensor models upon which the filter is based.

With these assumptions the standard Kalman filter measurement update process is given by:

$$\hat{\underline{x}}_T(t_i^+) = \hat{\underline{x}}_T(t_i^-) + \underline{K}(t_i)[\underline{z}(t_i) - \underline{H}_T(t_i)\hat{\underline{x}}_T(t_i^-)] \qquad (3\text{-}9)$$

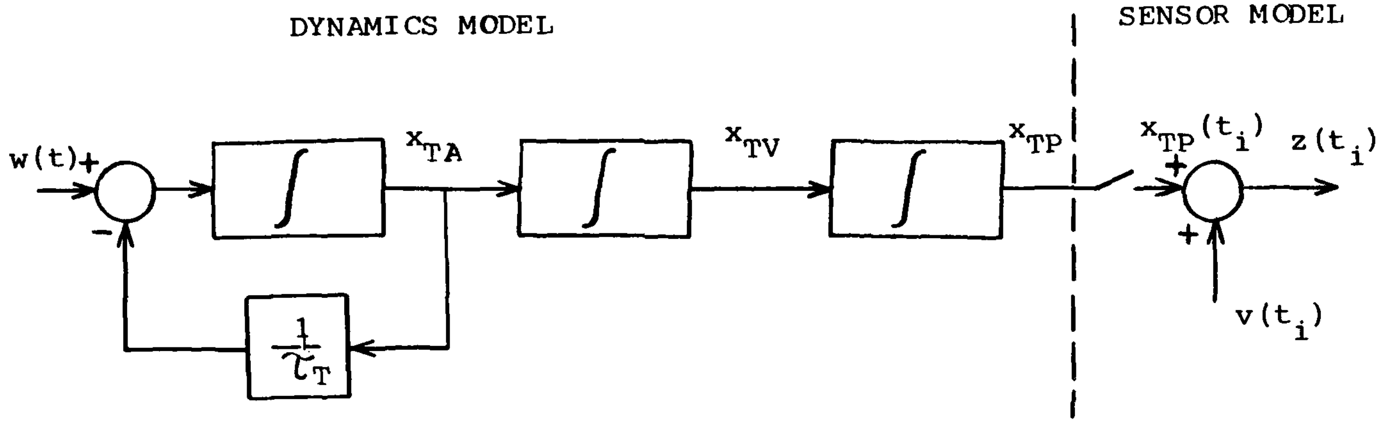


Figure 3.3 Target Dynamics and Sensor Models

$$\underline{P}_T(t_i^+) = \underline{P}_T(t_i^-) - \underline{K}(t_i)\underline{H}_T(t_i)\underline{P}_T(t_i^-)] \qquad (3\text{-}10)$$

$$\underline{K}(t_i) = \underline{P}_T(t_i^-)\underline{H}_T^T(t_i)\ [\underline{H}_T(t_i)\underline{P}_T(t_i^-)\ \underline{H}_T^T(t_i) + \underline{R}(t_i)]^{-1} \qquad (3\text{-}11)$$

with initial conditions $\hat{\underline{x}}(t_0) = \hat{\underline{x}}_0$ and $\underline{P}(t_0) = \underline{P}_0$. Figure 3.4 is the Kalman filter block diagram. Note that there is no feedback to the target, and so there is no feedback control term in Equations (3-5) or (3-9), or in the filter diagram. The state estimates out of this filter provide the target state estimates used by the controller.

Before we incorporate the target model and sensor model into the Kalman filter, we should verify that the resulting filter is stable. Therefore, we must examine the filter model's stabilizability with respect to the dynamics driving noise input w and detectability with respect to the measurement z output of the sensor system [12:464-465; 16:243-245]. By stabilizability, we mean the filter's unstable subspace is contained in its controllable subspace [12:462]. By detectability we mean, the filter's unreconstructible subspace is contained in its stable subspace [12:465]. As noted by Maybeck [18:92-93]: "By its definition, stabilizability is implied by either complete controllability or asymptotic stability... It can be readily be seen that either complete observability or asymptotic stability will imply detectability." Since the

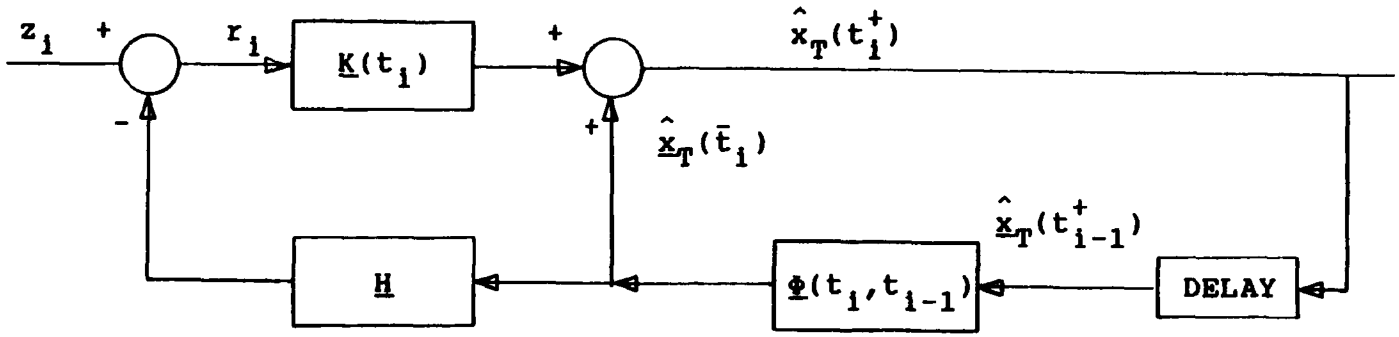

z_i
+
-
r_i
$\underline{K}(t_i)$
+
+
$\hat{x}_T(t_i^+)$
$\hat{\underline{x}}_T(\bar{t}_i)$
$\hat{\underline{x}}_T(t_{i-1}^+)$
$\underline{H}$
$\underline{\Phi}(t_i,t_{i-1})$
DELAY

Figure 3.4 Target Kalman Filter Block Diagram

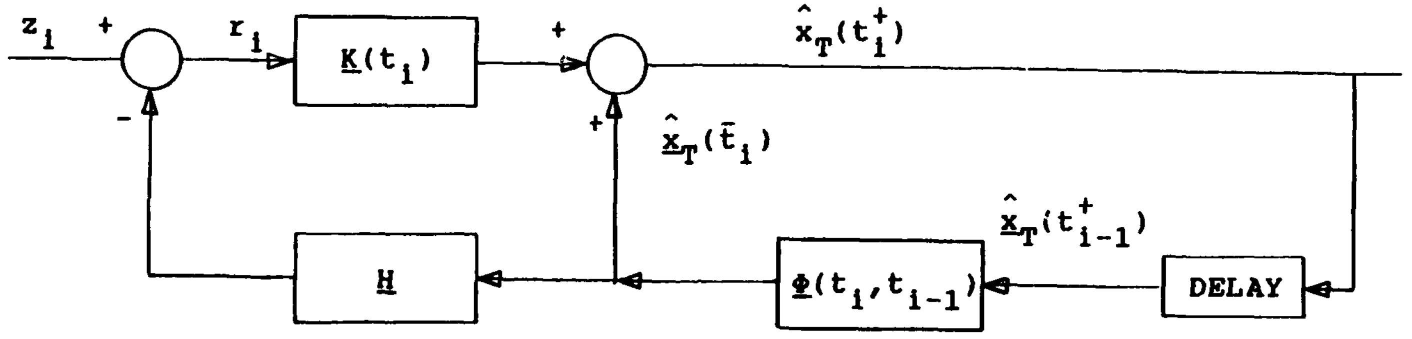

z_i
+
r_i
$\underline{K}(t_i)$
+
$\hat{x}_T(t_i^+)$
-
+
$\hat{\underline{x}}_T(t_i^-)$
$\hat{\underline{x}}_T(t_{i-1}^+)$
$\underline{H}$
$\underline{\Phi}(t_i,t_{i-1})$
DELAY

Figure 3.4 Target Kalman Filter Block Diagram

conditions for controllability and observability are stricter than stabilizability and detectability, establishing controllability and observability will establish the stability of the Kalman filter. For linear systems, the conditions for asymptotic stability are indistinguishable from conditions for bounded input-bounded output (BIBO) stability. Thus, showing that either set of conditions is satisfied will indicate that the filter is asymptotically, globally stable [16].

The system model described in Equation (3-4) is said to be stochastically controllable if there exists positive numbers α and β, $0 < \alpha < \beta < \infty$, and a positive integer N such that, for all $i \geq N$,

$$\alpha \underline{I} \leq \sum_{j=i-N+1}^{i} \underline{\Phi}(t_i,t_j)\underline{G}_d(t_{j-1})\underline{Q}_d(t_{j-1})\underline{G}_d^T(t_{j-1})\underline{\Phi}^T(t_i,t_j) \leq \beta \underline{I} \qquad (3\text{-}12)$$

The system model is said to be stochastically observable if there exists positive numbers α and β, $0 < \alpha < \beta < \infty$, and a positive integer N such that, for all $i \geq N$,

$$\alpha \underline{I} \leq \sum_{j=i-N+1}^{i} \underline{\Phi}^T(t_j,t_i)\underline{H}^T(t_j)\underline{R}^{-1}(t_j)\underline{H}(t_j)\underline{\Phi}(t_j,t_i) \leq \beta \underline{I} \qquad (3\text{-}13)$$

The chosen models do result in observable and controllable system models. Therefore, for our choice of dynamics and sensor models, the target filter is guaranteed to be stable. In a similar manner, we will next chose and verify the model to be used in the controller development.

3.2 Controller Gain Via LQ Synthesis.

With the controller structure specified as in Figure 3.1 (more will be said about this in Section 3.3), we need a method of determining the appropriate feedback gains. In this study we will use LQ synthesis techniques to generate the gain appropriate to a deterministic full-state feedback controller. Then, by applying assumed certainity equivalence we will use these gains for the NPB controller. Describing LQ synthesis will involve explaining the quadratic cost function. The solution to the minimization of the quadratic cost function results in a method of determining the optimal gain. Since the cost function and therefore the gain involve choosing weighting matrices, we will present Implicit Model following as an aid to determining appropriate weighting matrices. This section will discuss the use of the quadratic cost to aid the determination of controller gains.

3.2.1 Description of Quadratic Cost Function.

The purpose of the cost function is to represent what we want to happen to the system when it deviates from our desired response, or how we want the system to be controlled, so an approach to controller design can be developed. To see how the system model is incorporated into the weighting matrices of the cost function, consider the system described by the stochastic differential equation,

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) + \underline{G}(t)\underline{w}(t) \qquad (3\text{-}14)$$

or more precisely,

$$d\underline{x}(t) = \underline{F}(t)\underline{x}(t)dt + \underline{B}(t)\underline{u}(t)dt + \underline{G}(t)d\underline{\beta}(t) \qquad (3\text{-}15)$$

where:

$\underline{\beta}$ is a Brownian motion process with statistics

$$E\left\{ d\underline{\beta}(t) \right\} = \underline{0} \qquad (3\text{-}16a)$$

$$E\left\{ d\underline{\beta}(t)\, d\underline{\beta}^T(t') \right\} = \begin{cases} \underline{Q}(t) & t=t' \\ \underline{0} & \text{otherwise} \end{cases} \qquad (3\text{-}16b)$$

The solution to Equations (3-14) or (3-15) is:

$$\underline{x}(t) = \underline{\Phi}(t,t_0)\underline{x}(t_0) + \int_{t_0}^{t} \underline{\Phi}(t,\alpha)\underline{B}(\alpha)\underline{u}(\alpha)d\alpha$$

$$+ \int_{t_0}^{t} \underline{\Phi}(t,\alpha)\, \underline{G}(\alpha)d\underline{\beta}(\alpha) \qquad (3\text{-}17)$$

and the equivalent discrete-time system is described by the stochastic difference equation:

$$\underline{x}(t_{i+1}) = \underline{\Phi}(t_{i+1},t_i)\underline{x}(t_i) + \underline{B}_d(t_i)\underline{u}(t_i) + \underline{G}_d(t_i)\underline{w}_d(t_i) \qquad (3\text{-}18)$$

where:

$\underline{\Phi}$ is the state transition matrix associated with $\underline{F}$

$\underline{B}_d$ is the discrete time input distribution matrix given by:

$$\int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},\alpha)\underline{B}(\alpha)d\alpha \qquad (3\text{-}19)$$

and $\underline{u}(\alpha)$ is assumed to equal $u(t_i)$ for all α in the interval $[t_i,t_{i+1})$

$\underline{G}_d$ is the discrete time driving noise distribution matrix,

equal to the identity matrix if $\underline{Q}$ is evaluated as done below

$\underline{w}_d$ is an equivalent white noise with statistics:

$$E\left\{ \underline{w}_d \right\} = \underline{0} \qquad (3\text{-}20a)$$

$$E\left\{ [\underline{w}_d(t_i)\ \underline{w}_d^T(t_j)] \right\} = \begin{cases} \underline{Q}_d(t_i) & t_i=t_j \\ \underline{0} & \text{otherwise} \end{cases} \qquad (3\text{-}20b)$$

where

$$\int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},\alpha)\underline{G}(\alpha)\underline{Q}(\alpha)\underline{G}^T(\alpha)\underline{\Phi}^T(t_{i+1},\alpha)\ d\alpha \qquad (3\text{-}21)$$

Using the LQG assumptions and applying assumed certainty equivalence, we will use LQ synthesis techniques to determine the controller gains. Therefore we can drop the stochastic term in the subsequent development. One approach to representing the system in a cost function is applying quadratic penalties to deviations in the states and controls from zero. Then the cost function has the following form for a sample data controller [18:73]:

$$J = E\Big(1/2\underline{x}^T(t_{N+1})\underline{X}_F\underline{x}(t_{N+1}) + \sum_{i=0}^{N} 1/2[\underline{x}^T(t_i)\underline{X}(t_i)\underline{x}(t_i) + \underline{u}^T(t_i)\underline{U}(t_i)\underline{u}(t_i) + 2\underline{x}^T(t_i)\underline{S}(t_i)\underline{u}(t_i)] \Big) \qquad (3\text{-}22a)$$

or

$$J = E\Big(1/2\underline{x}^T(t_{N+1})\underline{X}_F\underline{x}(t_{N+1}) + \sum_{i=0}^{N} 1/2 \begin{bmatrix} \underline{x}(t_i) \\ \underline{u}(t_i) \end{bmatrix}^T \begin{bmatrix} \underline{X}(t_i) & \underline{S}(t_i) \\ \underline{S}^T(t_i) & \underline{U}(t_i) \end{bmatrix} \begin{bmatrix} \underline{x}(t_i) \\ \underline{u}(t_i) \end{bmatrix} \qquad (3\text{-}22b)$$

where:

E is the expectation operator

t_0 is the initial time

t_{N+1} is the final time

t_N is the last time control is computed and applied as a constant over the ensuing sample period

N is the number of sample periods from t_0 to the final time of control application t_N

$\underline{x}$ is the controller state vector.

$\underline{u}$ is the control input vector.

$\underline{X}_F$, $\underline{X}$, $\underline{U}$, and $\underline{S}$ are weighting matrices described below.

$\underline{X}_F$ and $\underline{X}$ are symmetric positive semidefinite weighting matrices used to emphasize the importance of minimizing each system state deviation from zero (perturbation states are driven towards zero) at the final time and each incremental time, respectively. A large value for a diagonal element of $\underline{X}$, as compared to other diagonal elements, would result in the design of controller gains that would maintain the system state associated with that diagonal element closer to zero than the other system states.

$\underline{U}$ is a symmetric positive definite matrix used to weight the importance of minimizing control application. A smaller value for a diagonal element of $\underline{U}$, as compared to other diagonal elements, would result in controller gains that would allow more energy to be used in the actuator associated with this diagonal element than allowed through other actuators. The choice of elements in $\underline{X}$ and the corresponding entries in $\underline{U}$ forms a tradeoff between the desire for minimum rms system states and minimum control energy application.

$\underline{S}$ is a matrix that arises naturally from a continuous-time cost function. Given a system described by:

$$\dot{\underline{x}}(t) = \underline{F}(t)\underline{x}(t) + \underline{B}(t)\underline{u}(t) \qquad (3\text{-}23)$$

with a continuous time quadratic cost of:

$$J_C = E\ 1/2\,\underline{x}^T(t_{N+1})\underline{X}_F\underline{x}(t_{N+1}) + \int_{t_0}^{t_{N+1}} 1/2 \begin{bmatrix}\underline{x}(t)\\ \underline{u}(t)\end{bmatrix}^T \begin{bmatrix}\underline{W}_{xx}(t) & \underline{W}_{xu}(t)\\ \underline{W}_{xu}^T(t) & \underline{W}_{uu}(t)\end{bmatrix} \begin{bmatrix}\underline{x}(t)\\ \underline{u}(t)\end{bmatrix} dt \qquad (3\text{-}24)$$

where $\underline{W}_{xx}$ is a positive semidefinite state weighting matrix, $\underline{W}_{uu}$ is a positive definite control input weighting matrix and $\underline{W}_{xu}$ is a cross weighting matrix. Note that these are continuous time weighting matrices. It can be shown [18:73-75] that the weighting matrices for the equivalent discrete time system are given by,

$$\underline{X}(t_i) = \int_{t_i}^{t_{i+1}} [\ \underline{\Phi}^T(t,t_i)\underline{W}_{xx}(t)\underline{\Phi}(t,t_i)]dt \tag{3-25}$$

$$\begin{aligned}\underline{U}(t_i) = \int_{t_i}^{t_{i+1}} [\ &\underline{\bar{B}}^T(t,t_i)\underline{W}_{xx}(t)\underline{\bar{B}}(t,t_i) + \underline{W}_{uu}(t) \\ &+ \underline{\bar{B}}^T(t,t_i)\underline{W}_{xu}(t) + \underline{W}_{xu}^T(t)\underline{\bar{B}}(t,t_i)\ ]dt\end{aligned} \tag{3-26}$$

$$\underline{S}(t_i) = \int_{t_i}^{t_{i+1}} [\ \underline{\Phi}^T(t,t_i)\underline{W}_{xx}(t)\underline{\bar{B}}(t,t_i) + \underline{\Phi}(t,t_i)\underline{W}_{xu}(t)]dt \tag{3-27}$$

where

$$\underline{\bar{B}}(t,t_i) = \int_{t_i}^{t} \underline{\Phi}(t,\alpha)\underline{B}(\alpha)d\alpha \tag{3-28}$$

Thus, the $\underline{S}$ terms arises from two sources. The first is a desire for cross terms in the discrete time representation, i. e. nonzero $\underline{W}_{xu}(t)$ terms above, or for placing quadratic penalty on deviations of $\underline{\dot{x}}(t)$ from zero (See Equation (3-23)). The second is a desire to exert control influence on the states over the entire sample period, i. e., the part of Equation (3-27) other than that due to nonzero $\underline{W}_{xu}(t)$. Also $\underline{S}$ is chosen so:

$$\begin{bmatrix} \underline{X}(t_i) & \underline{S}(t_i) \\ \underline{S}^T(t_i) & \underline{U}(t_i) \end{bmatrix}$$

is positive semidefinite for all t_i. This is done to ensure that the solution to the discrete backward Riccati recursion converges to a constant steady state solution [18:95].

Use of a quadratic cost for the control state terms ($\underline{x}$) in Equations (3-22a), (3-22b) and (3-24) arises from the desire to minimize the mean square error. The square error is useful as a cost basis as it provides for very large increases in cost as deviations in the control state from zero grow, while allowing for smaller increases in cost as deviations in the control state remain near zero. Use of the square of the control states also prevents any deviations to result in a lower or negative cost. Quadratics are also appropriate for the control energy.

Use of a quadratic cost for the control energy terms ($\underline{u}$) in Equations (3-22a), (3-22b), and (3-24) arises naturally from the common form of energy equations. Energy equations come in the form 1/2 mv^2 for a mass with translational velocity v or 1/2 Kx^2 for a spring with displacement x or 1/2 Cv^2 for a capacitor with voltage v or 1/2Li^2 for an inductor with current i [3:31;18]. Thus, a general control energy can be expressed in terms of a coefficient times the control variable squared.

As Maybeck describes the use of this cost function with linear systems and Gaussian noise models [18:12]:

> Although other types of cost functions are useful, quadratics are a good physical description of many control objectives, such as minimizing mean square errors while not expending inordinate amounts of control energy. Moreover, if one views most linear systems as being derived as linear perturbation models, by developing a controller

> that specifically attempts to maintain small values of quadratics in state and control deviations, one is inherently enhancing the adequacy of the linear perturbation model itself. However, perhaps the most important reason is that this combination of modelling assumptions yields a tractable problem whose solution is in the form of readily synthesized, efficiently implemented, feedback control law.

Once the quadratic cost function is established, we will seek the solution that results in the minimum cost. The next section will present the solution and show how the optimal gain is determined from this solution.

3.2.2 Controller Gain Synthesis. Equipped with Linear models and a Quadratic cost function we can synthesize an optimal linear full-state feedback control law. As an example, this will be done for a Proportional plus Integral (PI) controller. Details regarding the PI controller are in Sections 3.3.3 and 3.3.4. The gain for a proportional gain controller (regulator) is determined in the same manner.

First we consider an augmented system (see Section 3.3.3 for further discussion of the augmented system) described by the state equation which is both controllable and observable:

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a(t_{i+1},t_i)\,\underline{x}_a(t_i) + \underline{B}_{da}(t_i)\underline{u}(t_i) \qquad (3\text{-}29)$$

The optimal control law is [18:69]:

$$\underline{u}^*(t_i) = -\underline{G}_c^*(t_i)\underline{x}_a(t_i) + \underline{E}(t_i)\,\underline{y}_d \qquad (3\text{-}30a)$$

or

$$= -[\underline{G}^*_{c1}(t_i) \quad G^*_{c2}(t_i)] \begin{bmatrix} \underline{x}(t_i) \\ \hline q(t_i) \end{bmatrix} + \underline{E}(t_i)\, y_d \qquad (3\text{-}30b)$$

where

$\underline{G}^*_c(t_i)$ is the optimal controller gain

q is a pseudointegral state described in Sections 3.3.3 and 3.3.4

y_d is the nonzero setpoint

$E(t_i)$ is the gain applied to achieve the desired nonzero setpoint

Defining optimality as minimizing the quadratic cost function presented in the last section, then that minimum cost is given by:

$$1/2[\underline{x}^T(t_i)\underline{K}_c(t_i)\underline{x}(t_i)] \qquad (3\text{-}31)$$

where $\underline{K}_c(t_i)$ is the solution to the backward Riccati equation for the augmented state vector $[\underline{x}^T \ q]^T$:

$$\begin{aligned} \underline{K}_c(t_i) = {} & [\underline{\Phi}(t_{i+1},t_i) - \underline{B}_d(t_i)\, \underline{G}^*_c(t_i)]^T\, \underline{K}_c(t_{i+1}) \\ & [\underline{\Phi}(t_{i+1},t_i) - \underline{B}_d(t_i)\, \underline{G}^*_c(t_i)] + \underline{G}^*_c(t_i)\, \underline{U}(t_i)\, \underline{G}^*_c(t_i) \\ & + \underline{X}(t_i) - \underline{S}(t_i)\, \underline{G}^*_c(t_i) - \underline{G}^*_c(t_i)^T\, \underline{S}(t_i)^T \end{aligned} \qquad (3\text{-}32)$$

and

$$\underline{K}_c(t_{N-1}) = \underline{X}_F \qquad (3\text{-}33)$$

where $\underline{\Phi}$ is the state transition matrix of the controller-assumed augmented model for gain calculations and $\underline{B}_d$ is the corresponding control distribution matrix. The optimal controller gain is given by:

$$\underline{G}_c^*(t_i) = [\underline{U}(t_i)+\underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{B}_d(t_i)]^{-1}$$
$$[\underline{B}_d^T(t_i)\underline{K}_c(t_{i+1})\underline{\Phi}(t_{i+1},t_i)+\underline{S}^T(t_i)] \qquad (3\text{-}34)$$

We will seek to develop a constant gain controller, as this will be the least computationally burdensome and it will not impose unacceptable performance degradation. To accomplish this, the weighting matrices $\underline{X}$, $\underline{U}$, and $\underline{S}$ will be assumed to be time invariant, and the state transition matrix $\underline{\Phi}(t_{i+1},t_i)$ and control distribution matrix $\underline{B}_d$ are also time invariant. In addition, by letting N approach infinity in Equations (3-22a) and 3-22b), the effect of the cost at the time t_{N+1} on the total cost is assumed to be negligible. In other words, the terminal transient in the Riccati equation is short compared the the time for the entire process for this application. This reduces the equations to:

$$\underline{u}^*(t_i) = -\underline{G}_c^*\underline{x}(t_i) \qquad (3\text{-}35)$$

$$\underline{G}_c^* = (\underline{U}+\underline{B}_d^T\underline{K}_c\underline{B}_d)^{-1}\ (\underline{B}_d^T\underline{K}_c\underline{\Phi}+\underline{S}^T) \qquad (3\text{-}36)$$

where $\underline{K}_c$ is the solution to the steady-state backward Riccati equation [18:72-76],

$$\underline{K}_c = [\underline{\Phi}-\underline{B}_d\underline{G}_c^*]^T\underline{K}_c[\underline{\Phi}-\underline{B}_d\underline{G}_c^*]+\underline{G}_c^*\underline{U}\underline{G}_c^*+\underline{X}-\underline{S}\underline{G}_c^*-\underline{G}_c^{*T}\underline{S}^T \qquad (3\text{-}37)$$

Similarly, a constant gain E for Equation (3-30) is found to be:

$$E = [\underline{G}^{*}_{c1} - G^{*}_{c2}\ \underline{K}^{T}_{c12}/K_{c22}]\underline{\pi}_{12} + \pi_{22} \tag{3-38}$$

where

$$\underline{\pi} = \begin{bmatrix} \underline{\pi}_{11} & \underline{\pi}_{12} \\ \underline{\pi}_{21} & \pi_{22} \end{bmatrix} = \left[\begin{array}{c:c} \underline{\Phi} - \underline{I} & \underline{B}_d \\ \hdashline \underline{C} & \underline{0} \end{array}\right]^{-1} \tag{3-39}$$

where the elements of $\underline{K}_c$ are defined as:

$$\underline{K}_c = \begin{bmatrix} \underline{K}_{c11} & \underline{K}_{c12} \\ \underline{K}^{T}_{c12} & \underline{K}_{c22} \end{bmatrix} = \left[\begin{array}{cccc:c} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} \\ \hdashline k_{51} & k_{52} & k_{53} & k_{54} & k_{55} \end{array}\right] \tag{3-40}$$

Thus we have a method of determining the controller design based on the weighting matrices we choose. In the next section, the use of Implicit Model Following is used to determine a set of weighting matrices for use in the quadratic cost function.

3.2.3 <u>Weighting Matrices Using Implicit Model Following</u>. Model following in general is a means of forcing the outputs of a controlled system to behave as, or mimic, a desired system model that meets specifications [22:Sec II,2]. There are two types of model following controllers: explicit model followers and implicit model followers [22:Sec II,2]. This section will briefly describe the explicit model follower (as it is not used in this thesis) and then describe the implicit model follower. It is the implicit model follower that is used as an aid to determining the weighting matrices in this effort. These weighting matrices form the cost function and are used to determine controller gains via the backward Riccati equation (Equations (3-32), (3-33), and (3-34), or (3-36) and (3-37) for the steady state constant gain controller).

3.2.3.1 Explicit Model Following. In explicit model following, the desired model is explicitly simulated in the controller. The outputs of the desired model are multiplied by feedforward gains that have been generated by the controller. These control inputs drive the system to respond as the explicit model. Thus, the explicit model follower requires additional online computer processing to provide the desired model outputs. In general, the desired model is composed of simple first or second order systems for each response mode of interest, so the computational burden may be acceptable. Except that the target might be loosely considered an explicit model, this study does not incorporate an explicit model. Therefore, explicit model following will not be developed further.

3.2.3.2 Implicit Model Following. In Implicit Model Following, the desired model dynamics are introduced into the controller gain synthesis through the cost function via the weighting matrices. There are no model states to follow, so model following is only implied, hence the name implicit model following. The controller designed through implicit model following provides feedback whenever the system dynamic response deviates from the desired model dynamic response. As such, implicit model following attempts to match the system's closed-loop poles with those of the desired model. This ties the closed-loop system characteristics, including stability robustness, to the desired model characteristics. [18:Sec II,8]. As Miller points out [22:Sec V,15], there is a limit to the number of eigenvalues and eigenvectors that can be independently assigned in the closed-loop system by implicit model following. This limit is defined by the number of independent outputs and controls of the system. In Miller's words [22:Sec V,15]:

> If the controlled system has r independent controls and p independent outputs available for feedback, then at most max(r,p) eigenvalues may be assigned and min(r,p) entries of max(r,p) closed-loop eigenvectors may be assigned.

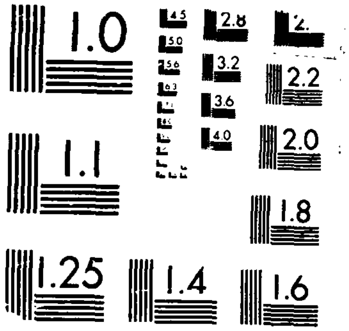
For our NPB controller, we have only one independent control and one independent output. Thus, only one eigenvalue and one eigenvector may be independently assigned in the closed-loop system. Through Implicit Model Following we can drive the system toward the desired pole. In Miller's words, referencing Kreindler and Rothschild [22:Sec II,6-7; 11:835-842]:

> The implicit formulation has an effect that is asymptotically (as state weighting grows large) equivalent to eigenstructure assignment, and the matching of the poles of the model and controlled system can produce desirable transient response. Rejection of unmodelled zero-mean disturbances may also be better than with an explicit controller, since good disturbance rejection characteristics can be included in the model response so that the disturbance states need not be modelled in the controller.

There are a number of ways to incorporate the desired model into the cost function. Maybeck, Miller, and Howey [15] presented a method of defining the discrete time weighting matrices from the continuous time dynamics model, continuous time weighting matrices and continuous time desired dynamics model. Johnson [9:173-180] developed an alternate approach that includes the state transition matrix. Since we are developing a discrete time controller using discrete time system models, we may desire to begin with discrete time weighting matrices. Therefore, we will present a method for determining the appropriate implicit model following weighting matrices from the discrete time system description and a set of discrete time weighting matrices.

To see how the desired model is incorporated into the weighting matrices of the cost function, consider the system described by Equations (3-14) to (3-21). Again, we will be using LQG assumptions and applying assumed certainty

AD-A194 622   TRACKING AND CONTROL OF A NEUTRAL PARTICLE BEAM USING   2/3
MULTIPLE MODEL ADAP..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI..
UNCLASSIFIED   L J HARAMBASIC DEC 87 AFIT/GE/ENG/87D-23   F/G 20/7   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

equivalence so that we can use LQ synthesis techniques to determine the controller gains. As before, we will drop the stochastic portion of the models to be used. The sample data equilvalent cost function without the incorporation of the implicit model is given by Equation (3-22a):

$$J = E\left\{ 1/2[\underline{x}(t_{N+1})^T \underline{X}_F(t_{N+1})[\underline{x}(t_{N+1})] + \sum_{i=0}^{N} [\ 1/2(\ \underline{x}^T(t_i)\underline{X}(t_i)\underline{x}(t_i) + \underline{u}^T(t_i)\underline{U}(t_i)\underline{u}(t_i) + 2\underline{x}^T(t_i)\underline{S}(t_i)\underline{u}(t_i)\ )\ ] \right\} \tag{3-22a}$$

Our real interest is in the continuous time outputs of the system is given by,

$$\underline{y}(t) = \underline{C}(t)\underline{x}(t) \tag{3-41}$$

where $\underline{C}(t)$ relates the controlled outputs to the system states. However, to the sample data controller the outputs of interest are.

$$\underline{y}(t_i) = \underline{C}(t_i)\underline{x}(t_i) \tag{3-42}$$

The objective is to force the output dynamics to match, as closely as possible, the output dynamics of the desired model, described by the equations:

$$\dot{\underline{y}}_m(t) = \underline{F}_m(t)\ \underline{y}_m(t) \tag{3-43}$$

or in the discrete time representation

$$\underline{y}_m(t_{i+1}) = \underline{\Phi}_m(t_{i+1},t_i)\underline{y}_m(t_i) \tag{3-44}$$

where $\underline{y}_m$ is of the same dimension as $\underline{y}$ and $\underline{\Phi}_m$ is the state transition matrix associated with the desired dynamics model $\underline{F}_m$. This model describes a closed-loop system which is primarily robust, but also has good performance characteristics. An appropriate $\underline{\Phi}_m$ can be determined from the desired eigenstructure, with poles placed in desired locations and eigenvectors as nearly orthogonal as possible for robustness [7].

To design a controller using a continuous cost function and implicit model-following, we would place a quadratic penalty on deviations of the continuous time output from the desired characteristics or $[\underline{y}(t) - \underline{F}\ (t)\underline{y}(t)]$. Analogously, we will place a quadratic penalty on $[\underline{y}(t_{i+1}) - \underline{\Phi}_m(t_{i+1},t_i)\ \underline{y}(t_i)]$ with a symmetric positive semidefinite output weighting matrix $\underline{Y}_I(t_i)$. The cost function now takes the form:

$$\begin{aligned} J = E\Big\{ & 1/2[[\underline{y}(t_{N+1})-\underline{\Phi}_m(t_{N+1},t_N)\underline{y}(t_N)]^T\ \underline{Y}_F(t_{N+1}) \\ & \qquad [\underline{y}(t_{N+1})-\underline{\Phi}_m(t_{N+1},t_N)\underline{y}(t_N)] \\ & + \sum_{i=0}^{N} [\ 1/2([\underline{y}(t_{i+1})-\underline{\Phi}_m(t_{i+1},t_i)\underline{y}(t_i)]^T\ \underline{Y}_I(t_i) \\ & \qquad [\underline{y}(t_{i+1})-\underline{\Phi}_m(t_{i+1},t_i)\underline{y}(t_i)] \\ & \qquad + \underline{u}^T(t_i)\underline{U}(t_i)\underline{u}(t_i) \\ & + 2[\underline{y}(t_{i+1})-\underline{\Phi}_m(t_{i+1},t_i)\underline{y}(t_i)]^T(t_i)\underline{S}(t_i)\underline{u}(t_i)\ ]\ \Big\} \end{aligned} \quad (3\text{-}45)$$

By applying Equations (3-23) and (3-42), the cost function can then be rewritten as:

$$J = E\left\{ 1/2[[\underline{y}(t_{N+1})-\underline{\Phi}_m(t_{N+1},t_N)\underline{y}(t_N)]^T \underline{Y}_F(t_{N+1}) [\underline{y}(t_{N+1})-\underline{\Phi}_m(t_{N+1},t_N)\underline{y}(t_N)] + \sum_{i=0}^{N} [\, 1/2(\, \underline{x}^T(t_i)\underline{X}_I(t_i)\underline{x}(t_i) + \underline{u}^T(t_i)\underline{U}_I(t_i)\underline{u}(t_i)) + 2\underline{x}^T(t_i)\underline{S}_I(t_i)\underline{u}(t_i)\ ] \right\} \quad (3\text{-}46)$$

where (dropping time index for clarity):

$$\underline{X}_I = [\underline{C\Phi}-\underline{\Phi_m C}]^T\underline{Y}_I[\underline{C\Phi}-\underline{\Phi_m C}] \quad (3\text{-}47)$$

$$\underline{S}_I = [\underline{C\Phi}-\underline{\Phi_m C}]^T\underline{Y}_I\underline{CB_d} + [\underline{C\Phi}-\underline{\Phi_m C}]^T\underline{S} \quad (3\text{-}48)$$

$$\underline{U}_I = [\underline{CB_d}]^T\underline{Y}_I\underline{CB_d} + [\underline{CB_d}]^T\underline{S} + \underline{S}^T\underline{CB_d} + \underline{U} \quad (3\text{-}49)$$

where the I supscript indicates an implicit model following weighting matrix. Taking the cost function summation out to infinity for the steady state condition,

$$J = E\left\{ 1/2 \left(\sum_{i=0}^{\infty} \begin{bmatrix} \underline{x}(t_i) \\ \underline{u}(t_i) \end{bmatrix}^T \begin{bmatrix} \underline{X}_I(t_i) & \underline{S}_I(t_i) \\ \underline{S}_I^T(t_i) & \underline{U}_I(t_i) \end{bmatrix} \begin{bmatrix} \underline{x}(t_i) \\ \underline{u}(t_i) \end{bmatrix} \right) \right\} \quad (3\text{-}50)$$

Equations (3-46) and (3-50) are in the familiar cost function form, for which we can determine the controller gains, using the backward Riccati equation as in Section 3.2.

The form of the cost function shown allows the use of standard LQ synthesis techniques to develop an optimal feedback controller that tries to force the system to meet classical performance specifications embodied in the model output, $\underline{y}_m$. The extent to which the desired response is achievable is dependent on the degree of difference between the system's inherent dynamics, represented by $\underline{\Phi}$, and the desired response dynamics, represented by $\underline{\Phi}_m$ [22:Sec II,3-5] and how heavily $[\underline{y}(t_{i+1}) - \underline{\Phi}_m\underline{y}(t_i)]$ is weighted through $\underline{Y}_I$ Equations (3-47), and (3-48). The desired model, represented by $\underline{\Phi}_m$, is the goal, but it is the weighting on $\underline{Y}_I$ that determines how much control energy is applied to achieve this goal. This form also allows the experience gained in an iterative design process to be useful when applying Implicit Model Following techniques.

The choice of an implicit model must be made carefully. Since the feedback gain is a function of the difference of the system model and the desired model through the matrix $(\underline{C\Phi}-\underline{\Phi}_m\underline{C})$, the implicit model-follower is particularily sensitive to parameter variations and plant modelling inaccuracies [22:Sec II,7]. In this study we are already aware of a problem with the mismodelling of the beam time constant. Therefore, we should be concerned about using implicit model following. However, if elements in $\underline{\Phi}_m$ dominate elements in $\underline{\Phi}$, then for small variations in $\underline{\Phi}$, $(\underline{C\Phi}-\underline{\Phi}\ \underline{C})$ will remain essentially the same. Even so, the elements of $\underline{\Phi}_m$ should not be set so as to result in an unstable closed-loop system. As such, there may be limitations to the performance or robustness improvement possible when using implicit model following.

3.3 Controller Structure

As with the selection of the target model, we have a great deal of freedom in chosing a controller. In chosing a controller, we desire minimum transient times and minimum steady state errors for wide variations in the parameters of the closed-loop system. Also, we desire a computationally fast structure. These desires shape the complexity of the controller. This section presents the controller used in this thesis. Prior to discussing the controller structure, some of the basic controller building blocks: Proportional Gain (PG) regulator, PG tracker, PI regulator, PI tracker, and Multiple Model Adaptive Controller (MMAC) are presented. Then we will present the controller used in this study.

3.3.1 Proportional Gain Regulator. The purpose of the regulator is to drive the beam centroid to the center of the detector array. The discrete-time state equation of the controlled beam is given by:

$$x_B(t_{i+1}) = \Phi_B(t_{i+1},t_i)x_B(t_i) + B_d(t_i)u(t_i) + G_d(t_i)w_d(t_i) \tag{3-51}$$

The optimal control law is given by:

$$u^*(t_i) = -G^*(t_i)x_B(t_i) \tag{3-52}$$

The optimal control gain is calculated as is Section 3.2.2 with the gain E being ignored. Since we are interested in directing the beam onto the target, we will consider the tracker next.

3.3.2 Proportional Gain Tracker. Maybeck and others have demonstrated how LQG techniques can be used to develop trackers [6:114-122]. LQ synthesis techniques require that the model used to determine controller gains have controllability with respect to the control input u and observability with respect to the commanded output y_c. For the time-invariant case, the controller needs stabilizability with respect to the control feedback u and detectability with respect to extracting the controlled output y_c. This is sufficient to ensure a solution to the backward Riccati equation. Using LQG techniques and assumed certainty equivalence, we first consider the deterministic LQ optimal full-state feedback controller based on the beam and target models with the uncertainities removed,

$$\underline{x}(t_{i+1}) = \underline{\Phi}_C(t_{i+1}, t_i)\underline{x}(t_i) + \underline{B}_{dC}(t_i)u(t_i) \qquad (3\text{-}53)$$

where

$$\underline{\Phi}_C = \left[\begin{array}{c|c} \Phi_B & 0 \\ \hline \underline{0} & \underline{\Phi}_T \end{array}\right] \qquad (3\text{-}54)$$

and $\underline{B}$ = [B 0 0 0]T, with B given by Equation (3-19). The controller continuous-time dynamics model is described in terms of:

$$\underline{F}_C = \left[\begin{array}{c|c} F_B & 0 \\ \hline \underline{0} & \underline{F}_T \end{array}\right] \qquad (3\text{-}55)$$

For the tracker we desire to point the beam at the target. Another way of expressing this desire is to drive the difference between the beam output and the target postion to zero. Then deviations from zero can be considered as errors given by:

$$e(t) = x_B(t) - x_{TP}(t) \qquad (3\text{-}56)$$

where the target position is taken as the reference variable. These errors also define the output state of the controller.

$$y_C = C_C\underline{x} = [C_B \;\; \underline{C}_T]\underline{x}$$
$$y_C = [1 \;\; -1 \;\; 0 \;\; 0]\underline{x} \qquad (3\text{-}57)$$

therefore y_C is the variable we want regulated to zero.

Again, we need controlability or at least stabilizability with respect to the control input u and observability or detectability with respect to extracting the controlled output y_C. The beam state and the target postion state combine to form the controlled output, and detectability can be shown to be satisfied. However, as implied in Jamerson's target model, there is assumed to be no control inputs applied (by the controller) to the target. In essence, we are assuming that the presence of the beam on the target has no impact on the target dynamics. From linear control theory, all the states of the target are thus uncontrollable from the point of entry of u [3:444; 18:92].

In addition, the presence of the two direct integrations in the dynamics model indicate the target motion of the open loop system is astable. This is the same as having two poles at the origin in the s-plane or two poles at the unity value of real axis of the z-plane. Without controllability or stability, there is no guarantee of stabilizability from the point of entry of u, nor of a solution to the steady-state Riccati equation used to generate constant controller gains (see Section 3.2.2) from the quadratic cost function. This requires a small adjustment to the controller model for the target. By moving the two poles at the origin of the s-plane to the left by some small epsilon, the controller model is now stable, although still not controllable. Thus a solution to the steady-state Riccati equation is guaranteed. By making this epsilon small and at least an order of magnitude away from the time constants of the target and the beam, the dynamics of the model system remain dominated by the target and beam time constants. Therefore the gains derived from the solution to the Riccati equation should still be applicable to the filter and system dynamics equations involving pure integrations [8;9], though performance evaluations are critical, to determine the impact of this model alteration. Thus, the adjusted controller gain model is given by,

$$\underline{\mathbf{x}}(t_{i+1}) = \mathbf{\Phi}'(t_{i+1},t_i)\underline{\mathbf{x}}(t_i) + \mathbf{B}_{dC}(t_i)u(t_i) \qquad (3\text{-}58)$$

where

$$\underline{\Phi}' = \left[\begin{array}{c|c} \Phi_B & \underline{0} \\ \hline \underline{0} & \underline{\Phi}'_T \end{array}\right] \qquad (3\text{-}59)$$

and $\underline{\Phi}'$ is determined from the inverse Laplace transform of $[s\underline{I}-\underline{F}']$ and the adjusted $\underline{F}'$ is given by:

$$F'_T = \begin{bmatrix} -\varepsilon & 1 & 0 \\ 0 & -\varepsilon & 1 \\ 0 & 0 & -1/\tau_T \end{bmatrix} \qquad (3\text{-}60)$$

It is important to note that this adjustment is only applicable for the controller model. The filter and truth models remain unchanged. With this change we can use the techniques of Section 3.2.2 to determine the appropriate gains.

3.3.3 Proportional Plus Integral Regulator. The proportional gain regulator lacks type 1 characteristics. Without type 1 control the steady state (mean) error cannot be driven to zero when constant unmodelled disturbances are present. To achieve type 1 control we need to include integral action in the control loop. Just as a model of the beam is incorporated into the regulator synthesis model so the gains are appropriately determined, we also must incorporate a model of the integration process. This allows the appropriate feedback gains to be determined for each state and the integrator by LQ synthesis methodology, rather than a trial and error approach to determine the best gain settings. Since we are using a discrete system, it is appropriate to model the integration process as a summation process or pseudointegral [9:46;18]. Following Johnson's development, we include a non-zero setpoint control y_d. The nonzero setpoint control will allow pointing of the beam at an aimpoint different from the detector array centroid. The pseudointegral becomes [18:133]:

$$q(t_i) = q(t_0) + \sum_{j=0}^{i-1} [y_B(t_j) - y_d] \qquad (3\text{-}61a)$$

$$= q(t_{i-1}) + [x_B(t_{i-1}) - y_d] \qquad (3\text{-}61b)$$

Figure 3.5a illustrates the pseudointegral for the PI regulator. The pseudointegral is included in the controller model by augmenting Equation (3-61b) to controller model of Section 3.1.3, Equation (3-51). The augmented deterministic controller gain model becomes,

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a \underline{x}_a(t_i) + \underline{B}_{da} u(t_i) + \underline{D}_a y_d \qquad (3\text{-}62a)$$

$$\begin{bmatrix} x_B(t_{i+1}) \\ q(t_{i+1}) \end{bmatrix} = \begin{bmatrix} \Phi_B & 0 \\ C & 1 \end{bmatrix} \begin{bmatrix} x_B(t_i) \\ q(t_i) \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} u(t_i) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} y_d \qquad (3\text{-}62b)$$

The optimal control law is given by:

$$u^*(t_i) = -\underline{G}^*(t_i)\underline{x}_a(t_i) \qquad (3\text{-}63)$$

The optimal control gain is calculated as is Section 3.2.2.

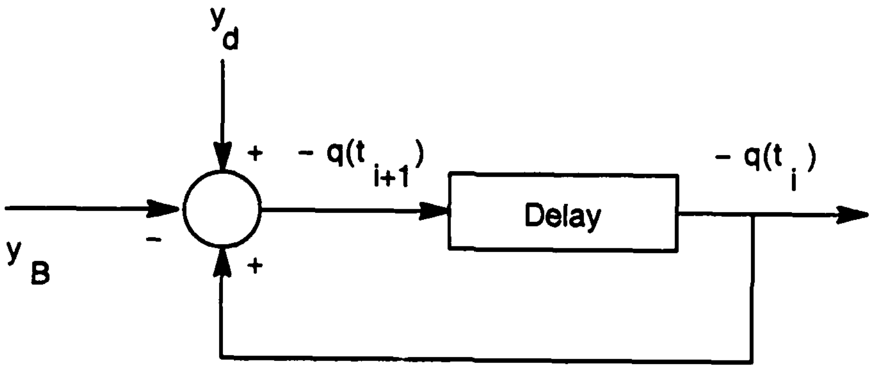


Figure 3.5a Pseudo-Integral for a Regulator

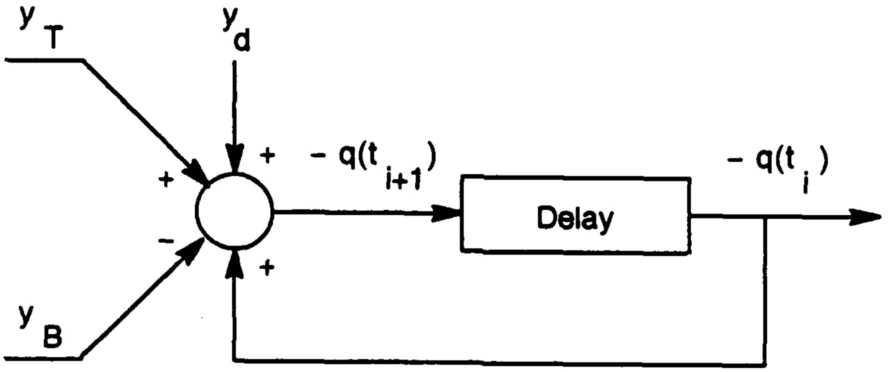


Figure 3.5b Pseudo-Integral for a Tracker

3.3.4 Proportional Plus Integral Tracker. As with the PI regulator, we also incorporate a model of the integration process in the PI tracker. This allows the appropriate feedback gains to be determined for each state and the integrator by LQ synthesis methodology, rather than a trial and error. As before, we are using a a summation process or pseudointegral as a model of the integration process. Again we include a non-zero setpoint control y_d. However, the nonzero setpoint control will now allow pointing of the beam at an aimpoint different from the target centroid. The pseudointegral then becomes [9:46;18:133],

$$q(t_i) = q(t_0) + \sum_{j=0}^{i-1} [y_B(t_j) - x_{TP}(t_j) - y_d] \qquad (3\text{-}64a)$$

$$= q(t_{i-1}) + [x_B(t_{i-1}) - x_{TP}(t_{i-1}) - y_d] \qquad (3\text{-}64b)$$

Figure 3.5b illustrates this pseudointegral. The pseudo-integral is included in the controller model by augmenting Equation (3-64b) to controller model of Section 3.1.3 Equation (3-58). The augmented deterministic controller gain model now becomes,

$$\underline{x}_a(t_{i+1}) = \underline{\Phi}_a \underline{x}_a(t_i) + \underline{B}_{da} u(t_i) + \underline{D}_a y_d \qquad (3\text{-}65a)$$

$$\begin{bmatrix} \underline{x}(t_{i+1}) \\ q(t_{i+1}) \end{bmatrix} = \begin{bmatrix} \underline{\Phi}' & 0 \\ \underline{C}_C & 1 \end{bmatrix} \begin{bmatrix} \underline{x}(t_i) \\ q(t_i) \end{bmatrix} + \begin{bmatrix} \underline{B}_d \\ 0 \end{bmatrix} u(t_i) + \begin{bmatrix} \underline{0} \\ -1 \end{bmatrix} y_d \qquad (3\text{-}65b)$$

Equation (3-63) can be written as (with time arguments dropped):

$$u^* = -[G_B^* \; \underline{G}_T^* \; G_q^*] \begin{bmatrix} x_B \\ \underline{x}_T \\ q \end{bmatrix} \qquad (3\text{-}66)$$

This is the form of the elemental controller used in this study. Figure 3.6 depicts this controller in block diagram form. Elemental controllers of this form will be used together in a multiple model structure described next.

3.3.5 Multiple Model Adaptive Controller. The multiple model adaptive controller (MMAC) was incorporated by Johnson to combat the problem sluggish response to highly dynamic targets. The derivation of the MMAC parallels that presented in Section 2.4 for multiple model adaptive estimation Equations (2-23) through (2-31). Analogous to that development, the feedback out of the MMAC is the weighted sum of the outputs of each elemental controller as given by:

$$u(t_i) = \sum_{k=1}^{K} u_k(t_i) \; p_k(t_i) \qquad (3\text{-}67)$$

$$u(t_i) = \sum_{k=1}^{K} -\underline{G}_{ka}(a_i) \; \underline{x}_{ka}(t_i) \; p_k(t_i) \qquad (3\text{-}68)$$

where here K is the number of elemental controllers and the

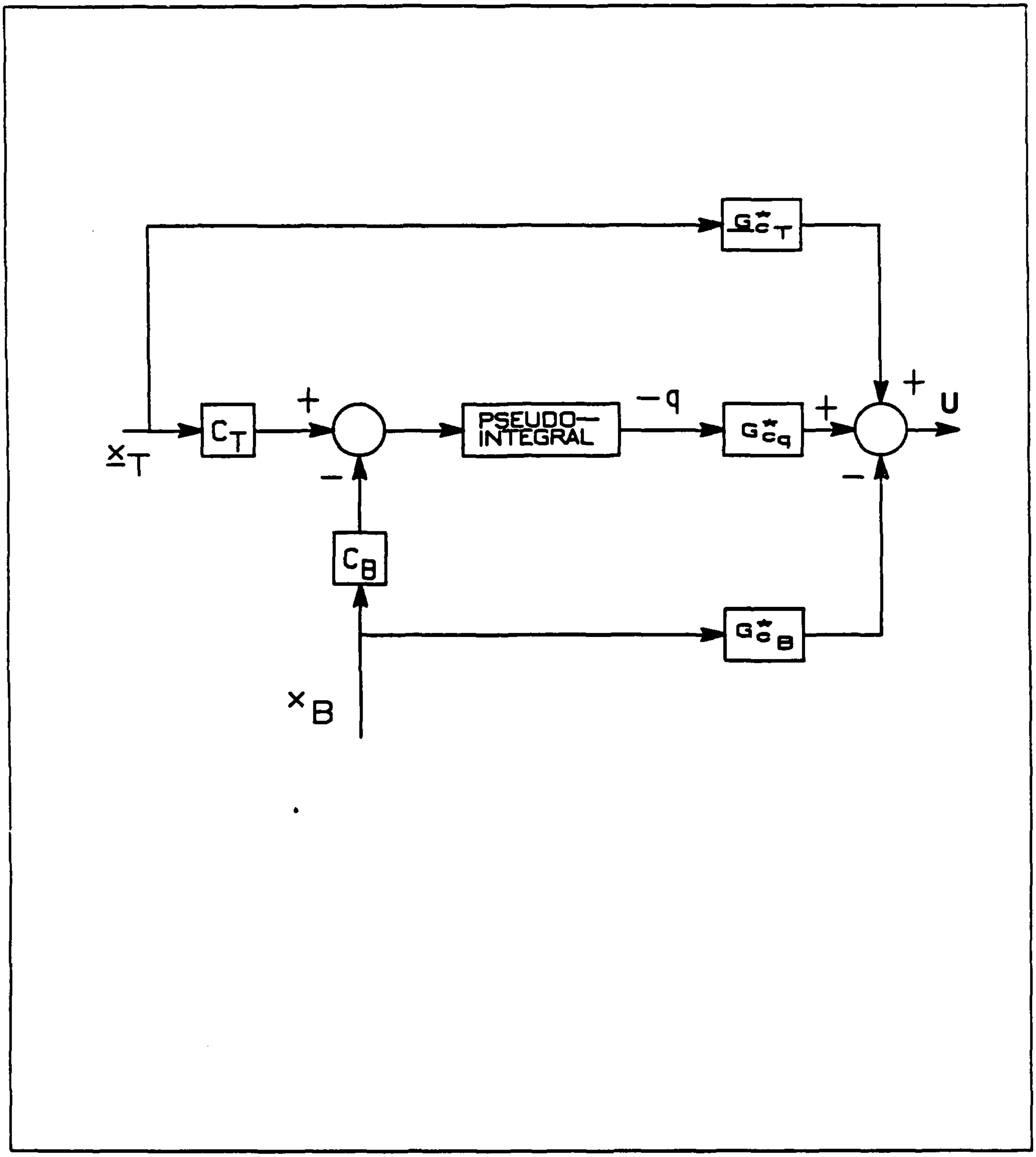


Figure 3.6 Elemental PI Controller Block Diagram

subscript "a" denotes "augmented". For Johnson's MMAC [9], each p_k was the probability weight associated with each Kalman filter. However, here $p_k(t_i)$ is probability of both the Meer filter estimate and the Kalman filter estimate given by:

$$p_k(t_i) = p_{Meer(m)}(t_i)\, p_{Kalman(\ell)}(t_i) \qquad (3\text{-}69)$$

$$\sum_{m=1}^{M} p_{Meer(m)} = 1 \qquad (3\text{-}70)$$

$$\sum_{\ell=1}^{L} p_{Kalman(\ell)} = 1 \qquad (3\text{-}71)$$

where here, M is the number of Meer filters and here, L is the number of Kalman filters. Thus, K=LM. Figure 3.7 depicts this MMAC.

3.3.6 Choosing a Controller. In previous thesis efforts, the controllers developed were: the Proportional Gain (PG) controller [32], the Proportional plus Integral (PI) controller [8; 23], and the multiple model adaptive controller (MMAC) composed of a bank of PI controllers [9]. Each of these controllers were constant gain devices. To check the feasibility of using the state estimates provided by the Meer filter in a controller design, the proportional gain controller was developed by Zicker [32]. This controller featured a PG target tracker designed around a PG

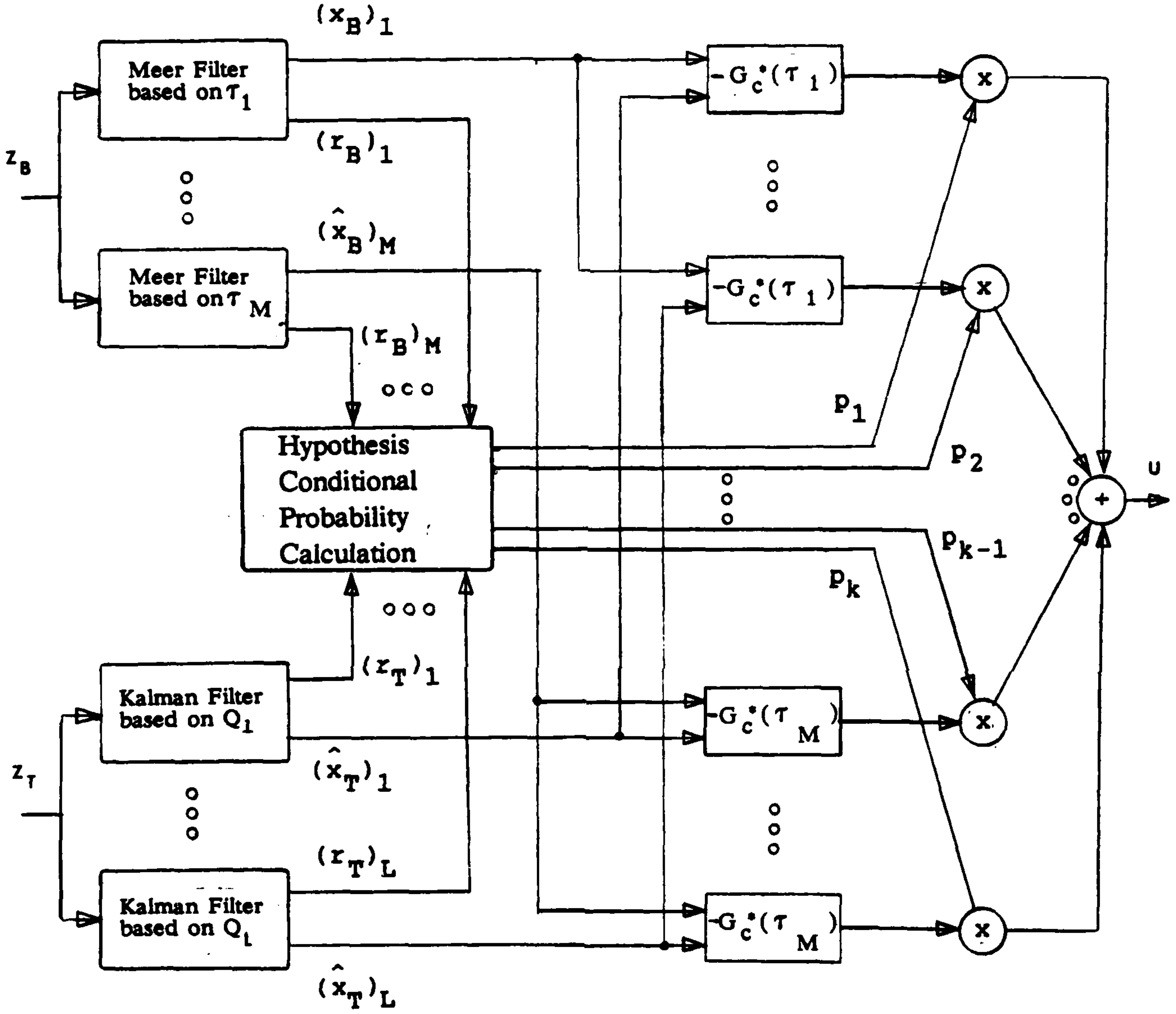


Figure 3.7 Multiple Model Adaptive Controller

beam regulator. With this design, Zicker showed that the use of the Meer filter in a controller design was feasible. However, the PG controller inherently lacks the type-one characteristics necessary to reject unmodelled constant disturbances injected into the target position state [3:196; 18:132]. As models can only approximate the real world, our system will always be subject to these unmodelled disturbances. Thus, Moose [23] began development of a controller based on a type-one Proportional plus Integral (PI) target tracker designed around a beam regulator. Jamerson continued Moose's development, but exchanged Moose's PI tracker with another PI tracker based on the more realistic target model that Jamerson incorporated into that study. These PI trackers were able to reject constant unmodelled disturbances (drive the mean error between the beam and the target positons to zero), as desired. However, they were slugglish and had difficulty tracking highly maneuverable targets [8; 9]. To improve performance against highly maneuverable targets, Johnson [9], developed a multiple model adaptive controller (MMAC) based on elemental controllers that used a PI target tracker. Each controller was based on a separate assumption about the approximate strength for the assumed target dynamics driving noise. Although variations in dynamic driving noise strength have no effect on the calculation of controller gains, the controller outputs differed due to the use of separate Kalman filter models of the target to supply the corresponding controllers with state estimates of the target. Thus, the controller gains in Figure 3.7 show no dependency on the individual Q value assumed by the target state Kalman filter. The differences in the assumed target dynamics driving noise strength does affect the state estimates generated by each Kalman filter. Although the MMAC performed well [9], Johnson found a severe robustness problem in the elemental PI controllers for a mismodelled beam time constant. One suggested solution to the robustness problem is the use of multiple model adaptive techniques on the beam time constant. That is the technique explored in this study.

Using multiple model techniques with the beam time constant results in a structure similar to Johnson's MMAC. There are some notable differences between the structures. Unlike an adaptation on the target driving noise strength, adapting on the beam time constant does result in different controller gains for each controller. Each controller then uses the beam estimate from its corresponding Meer filter, based on the same beam time constant, to generate the appropriate feedback. Each controller output is then weighted based on the residual performance of its target filter and its Meer filter to form the actual output. Thus, the controller structure for this study is a bank of nine PI controllers based on a unique hypothesis of the beam time constant and target driving noise strength. As in past three thesis efforts, each elemental PI controller is based on a PI tracker. The appropriate constant gains are generated through the use of LQ synthesis techniques based on the assumption of full-state feedback. Then through the use of assumed certainty equivalence, the controller constant gain blocks are provided the target and beam estimates from the respective Kalman and Meer filters. Once we have chosen a structure or our controller, we can begin the synthesis of the controller gains via LQG and Implicit model following techniques. Section 3.2 in fact discussed this synthesis in detail.

3.4 Summary

This chapter presented the current target model being used in this research as a third order Gauss-Markov position process. This satisfied the linear model and Gaussian noise requirements for a tractable control design. Next, the quadratic cost function for synthesizing each elemental controller gain was described. The quadratic cost function results in a tractable approach to controller design. Linear models and a quadratic cost criterion, allows the use of LQ synthesis techniques to determine controller gains. Assumed certainty equivalence is then used to incorporate these gains into the NPB controller. Weighting matrices in the cost function allow the user to specify how the control energy will be applied to regulate the states. Implicit model following techniques are one method of determining the appropriate weighting matrices for acheiving desired performance and robustness. The controller structure chosen for this study was a multiple model PI controller. The PI controller drives steady state mean errors to zero even in the presence of constant unmodelled disturbances. The multiple model approach attacks performance and robustness problems by processing a number of simple controllers in parallel, rather than increasing the complexity of a single controller. The next chapter discusses the analysis performed on the controller design of this chapter.

IV. SIMULATION DESIGN AND EVALUATION PROCEDURE

A design is not complete until its performance is validated against some pre-specified standard, baseline, goal, or specification. The first step in this validation process is an evaluation of performance using computer simulation. This chapter presents the method used to simulate the NPB tracking problem and evaluate estimator and controller performances in this study.

Two performance evaluations are made. The first evaluation is of the MMAE Meer filter's ability to estimate the position of the NPB centroid. The MMAE Meer filter is composed of a bank of Meer filters. Each Meer filter is based on a separate hypothesis about the value of the beam time constant. The second evaluation is of the ability of an MMAC based on an implicit model following to point the NPB at a given target. The MMAC is composed of a bank of nine PI controllers, each operating with a unique combination of one of three Kalman filters and one of three Meer filters. Each Kalman filter is based on a separate hypothesis about the strength of the target driving noise.

This chapter begins by presenting the analysis procedure. Following this, it describes the software tools used to produce and assimilate the data for performance evaluations. Finally, filter, controller, and simulation parameters used in this study are presented.

4.1 Performance Analyses

Eight different performance analyses are required to evaluate the MMAE Meer filter and to evaluate the use of implicit model following weighting matrices in the synthesis of controllers. The first five analyses involve the evaluation of the MMAE Meer filter. The last three analyses involve the MMAC using cost weighting matrices developed by implicit model following techniques.

The MMAE Meer filter is an adaptive filter composed of a bank of elemental Meer filters, each based on an assumed beam time constant τ_B. The Meer filter itself is an adaptive filter composed of elemental Synder-Fishman filters, each based on a different hypothesis about the sequence of noise and signal events. The effects of such a second level of adaptation are unknown. Therefore, an evaluation of the estimation and control performance with the MMAE Meer filter is necessary to determine whether this adaptation can adequately estimate the beam location for off-nominal beam time constants.

The MMAC is composed of a bank of elemental PI controllers. Each elemental controller uses one of three Kalman filters to estimate the target state (differing from each other on basis of dynamics driving noise strength) and one of three Meer filters (differing on assumed beam time constant) to estimate the beam state. Each PI controller will be based on a unique combination of the target dynamics driving noise strength [9] and beam time constant. Thus the bank will be composed of 9 elemental PI controllers. The analysis of the controller performance will allow selection of the weighting matrices (used for controller synthesis) that provide the best tradeoff of performance at design conditions versus robustness.

4.1.1 Evaluating the MMAE Meer Filter. Five analyses will be conducted to evaluate the performance of the MMAE Meer filter. The first analysis set will determine whether the use of an MMAE Meer filter can consistently provide precise estimates of the beam location when the beam time constant is uncertain. That a MMAE Meer filter provides good state estimates does not imply that it also provides good parameter estimates. The second analysis will determine whether the MMAE Meer filter can provide an adaptive estimate of the beam time constant while providing good state estimates. As such, this analysis is run concurrently with the first. The third analysis will justify the use of an elemental Meer filter depth of one (D = 1). The last two analyses will define the limits on the beam time constant for which the MMAE Meer filter will be able to operate.

Johnson experienced significant degradation in target tracking performance with only 2% variation in filter beam time constant from the true value [9:118]. Since he found little change in the controller gains for small changes in the beam time constant [9:124], we suspect that the Meer filter does not provide an adequate estimation of the position of the beam centroid position when the beam time constant is mismodeled. Thus the focal point for our first analysis is the MMAE Meer's ability to estimate NPB centroid location.

For the first analysis, the MMAE Meer filter consists of 3 Meer filters, based on the nominal beam time constant, nominal minus 4%, and nominal plus 4%, with a Meer filter memory depth of 3. The weighted beam position estimate of the filter is compared to the true beam position to evaluate true performance capability. This first analysis demonstrates whether a simple MMAE Meer filter can consistently provide accurate beam position estimates when the true beam filter time constant differs from the assumed time constants of the individual models. The true value of the beam time constant will be set to the nominal, at nominal plus 2%, and at nominal minus 2%. The first setting provides an optimal setting of the true beam time constant with respect to the filter-assumed values. The last two setting are to provide a worst case performance by mismatching the true value with the filter-assumed values as much as possible: no single elemental filter is specifically tuned for these parameter values. If performance is poor, then additional Meer filters may be required to provide a finer discretization of the beam time constant. This analysis will indicate whether the MMAE Meer filter is a viable method of solving the robustness problem found by Johnson with regard to the beam time constant [9]. Subsequent analyses will use an MMAE with the number of Meer filters determined as appropriate in this first analysis.

The second analysis determines whether the filter can provide an adaptive estimate of the beam time constant while providing good state estimates. A good adaptive beam time constant estimate could be used to propagate the beam estimate. A good estimate of the beam could also be used in adaptive controller designs that calculate the feedback gain in real time based on an estimate of the beam time constant.

The third analysis evaluates the performance of the MMAE Meer filter when the filter memory depth is decreased from D = 3 to D = 1. This will reduce the number of propagating elemental Snyder-Fishman filters in each elemental Meer filter from eight to two. Additionally, each elemental Meer filter can be expressed equivalently with only a single elemental Synder-Fishman filter, but with its gain expressed as a function of residual size, as seen in Section 2.5.2.2 [8; 9; 32]. Although this may slightly increase the average RMS error, it should significantly decrease the computational loading. It makes on-line feasibility much more reasonable, and so a tradeoff analysis is warranted.

The fourth MMAE Meer filter analysis will determine the effects of a linearly changing and sinusoidally varying true beam time constant on the performance of the MMAE Meer filter. This will give insight into the MMAE Meer filter's ability to track a real world particle beam with physically motivated parameter changes.

The final analysis will determine the limits on the variation of beam time constant using the MMAE Meer filter. Limits using additional Meer filters may be investigated. These limits would indicate the discretization necessary to cover the range of beam time constants over which the MMAE Meer filter may operate.

4.1.2 Evaluating the MMAC with Implicit Model Following Weighting Matrices.

In the first controller analysis, candidate weighting matrices are determined using implicit model following. The performance with these weighting matrices is measured to ensure reasonable performance at design conditions. The next two analyses indicate performance for variations in beam and target parameters with the weighting matrices that have reasonable performance at design conditions as determined in the first analysis. A robustness analysis is performed to indicate performance when controller-assumed parameters do not match actual beam and target parameter. This analysis will allow the selection of the weighting matrices that provide the best tradeoff in performance at design conditions versus robustness. A sensitivity analysis is performed to indicate performance capabilities when the controller is operated over a variety of design conditions. In a sensitivity analysis, the controller-assumed parameter matches the corresponding beam or target parameter. This will indicate the range of design conditions over which the weighting matrices are applicable.

The determination of the best cost weighting matrices based on implicit model following techniques is not an exact science. There may exist a number of sets of weighting matrices which provide an adequate level of performance with a reasonable degree of robustness. Therefore, a performance analysis is conducted to determine the weighting matrix set which provides good performance at design conditions and also enhances the MMAC's robustness. In other words, the goal is to select the set(s) of weighting matrices that best minimize the average RMS error (without using excessive amounts of control) when the truth model matches the filter model, while simultaneously providing at least stability and some degree of desirable performance when controller-assumed parameters do not match corresponding beam and target parameters.

A robustness analysis is performed to evaluate the MMAC controller's performance when a controller-assumed model parameter differs from that of the truth model. The robustness analysis is conducted under the same guidelines as the sensitivity analysis, but differs in that a parameter is changed in the truth model without informing the controller. The purpose of this study is to determine whether the use of implicit model following techniques in controller design, can result in the design of a robust controller while maintaining adequate performance at design conditions.

The baseline against which these analyses are compared is defined as a PI controller which receives the beam and target states from the truth model and has access to the truth model parameters. Such a baseline provides the best performance possible for the chosen set of controller gains. A second benchmark is defined as a PI controller which receives state estimates from a single Kalman filter and a single Meer filter which have access to the correct parameters. The second baseline provides a realistic assessment of the best performance possible from controllers that operate with state estimates rather than the actual states.

The purpose of the sensitivity analysis is to evaluate the performance of the MMAC controller based on the weighting matrices determined in the previous analysis as its design parameters (both truth and filter) are changed. Sensitivity to parameter variations may indicate the need for further adaptive estimation. Since the contoller gains are dependent on the design parameters, as well as choice of weighting matrices, robust performance at a given set of design parameters may not be indicative of robustness for other sets of design parameters.

The sensitivity analysis is based on the variation of a single parameter setting in both the controller-assumed model and truth model (i.e. "real world"), while other parameters remain at the nominal values presented in the previous sections. For this study, data will be collected as parameters are varied one order of magnitude above and below the nominal value. The analysis evaluates all the beam and target parameters except the filter depth, which is set to one (D = 1). These results are compared to the results obtained by Johnson [9:95] using the cost weighting matrices that are based on the results in Chapter 5 of Johnson's thesis. These weights are: U=1, X_{11}=100, X_{55}=10 and all other weights and cross weight equal to zero, where U is the weight on the control energy, X_{11} is the weight on tracking error and X_{55} is the weight on the pseudo-integration state (See Section 3.3.4).

4.1.3 <u>The Method - Monte Carlo Simulation</u>. It will be necessary to evaluate the MMAE Kalman filter and MMAE Meer filter state estimation error statistics, to ensure these filters are not diverging and are in fact providing viable estimates of important state variables. It will also be required to evaluate MMAC tracking errors. The performance of an estimator/controller can often be analyzed by either covariance analysis or Monte Carlo simulation [16:329]. The least time consuming method is covariance analysis, which requires only one software run to generate the time history of the estimation error covariance. Unfortunately, covariance analysis should not be applied to this system, as proper covariance analysis is limited to linear stochastic estimators and/or controllers that use prespecified measurement update times. The adaptive mechanisms in both the controller and the MMAE Meer filter, as well as the Meer filter itself, violate the linearity restriction, and the elemental Synder-Fishman filters have a varying sample rate that is not prespecifiable. Therefore, the more time consuming Monte Carlo simulation is used to evaluate the particle beam estimator/controller. Next, we describe the software used to facilitate the generation and collection of data for this study.

4.2 The Tools - SOFE and SOFEPL

The performance data are generated from a specially modified version of a software package called SOFE (Simulation for Optimal Filter Evaluation) [24]. Subsequent performance analysis is performed using SOFEPL (SOFE Plotter) [6]. SOFE is a general purpose Monte Carlo simulation program designed for evaluating systems that use Kalman or extended Kalman filters. This section begins by describing how SOFE was modified and used to simulate the NPB, target, Meer filter, Kalman target filter and the controller. Following this, the use of SOFEPL to assimilate the data into error statistics is presented.

4.2.1 Data Gathering with SOFE.

SOFE allows the user to simulate a truth model and a Kalman or an extended Kalman filter model that can be described by a set of stochastic differential equations. SOFE also allows the user to specify the measurement format of the filter update, and if desired, to include impulsive feedback control. However, the incorporation of the Meer filter required a modification to the basic SOFE code. In Kalman or extended Kalman filter applications, the update times occur at prespecifiable constant intervals, whereas the Meer filter updates occur at random intervals. Therefore, it was necessary for Meer to modify the SOFE software to allow propagation to the Meer filter update time. Therefore, this section will first describe the modifications necessary to include random interval updates. Then, the equations used to simulate the NPB and target models are presented. Following this, the routines necessary to simulate the Meer filter model are described. Then the Kalman filter model for the target is presented. Finally, the method used to provide continuous feedback (piecewise constant, as generated by a computer and passed through a zero order hold) with the MMAC controller is described. The SOFE source code that Meer altered and that is used throughout the research is the FORTRAN 4, May 1982 version of SOFE [21:79].

4.2.1.1 Random Interval Updating. Since the Kalman filter is based on a fixed sample period, SOFE updates the Kalman filter at regular time intervals. Thus, Meer [21] had to modify the SOFE source code to accommodate the random arrival time of Poisson distributed signal and noise events at the detector (See Section 2.2). Meer altered the time that a user-defined error statistics generation routine is called, from a constant interval to the computed time of the next signal or noise event, whichever is soonest to occur. For simplicity, the times to the next signal and noise events are calculated and compared in the simulation from user-defined signal and noise rate parameters. The equation used to calculate the varying sample period is derived from the Poisson process density function,

$$p(y) = -[[\hat{\lambda}\, t]^{y}/y!]\ \exp[-\hat{\lambda}\, t] \qquad (4\text{-}1)$$

where p(y) is the probability that y events occurred within time period t, and $\hat{\lambda}$ is the mean signal arrival rate or mean noise arrival rate. Because we are interested in generating a random sample period with the next event occurring at the completion of the sample period, y is set to zero (i.e., no events prior to the end of the sample period), and the random sample period, t, is calculated from the inverse mapping function of Equation (4-1) [2; 9:83-84; 29:62].

$$t = -[\hat{\lambda}]^{-1}\ \ln[p(0)] \qquad (4\text{-}2)$$

where p(0) is defined as the probability that t amount of time has passed before an event arrives, and $0! = 1$. This probability has a range of $0 \leq p(0) \leq 1$. Because we are interested in generating samples of t_1, $p(0)$ is selected randomly from the range of $p(0)$, and this process is used to simulate the arrival time of the next signal- or noise-induced event. Both the signal and noise are Poisson processes, but each has its own mean arrival rate and they are independent processes. Thus, modified SOFE simulates a varying sample period (the time to the next signal- or noise-induced event, whichever comes first) as a Poisson time process.

The mean signal arrival rate is defined as

$$\overline{\lambda}_s = \Gamma \, (2\pi R)^{1/2} = \bar{n}/(t_F - t_0) \tag{4-3}$$

where Γ is the maximum amplitude of the signal rate density function, R is the beam dispersion, and $\bar{n}$ is the number of signal-induced events expected over the duration of the simulation run, $(t_F - t_0)$. The relationship between the mean signal arrival rate and the expected signal rate parameter, $\lambda_s(t,r,x(t))$, can be shown by integrating Equation (2-7) (shown for the one dimensional case),

$$\overline{\lambda}_s(t) = \int_{-\infty}^{\infty} \lambda_s(t_1, \alpha, x(t)) d\alpha$$

$$= \int_{-\infty}^{\infty} \Gamma(t_1) \exp[-\alpha^T R^{-1}(t_1)\alpha/2] d\alpha \tag{4-4}$$

where $\alpha=[r-H(t)x(t)]$. For this study, we are assuming that Γ and R are constant. Although there is no known closed form expression for the integral of the Gaussian density function, we know that the entire area under a probability density function must have a magnitude of one,

$$\int_{-\infty}^{\infty} [(2\pi)^{m} \; R \;]^{-1/2} \; \exp(-\alpha^{T} R^{-1} \alpha/2)d\alpha = 1 \qquad (4\text{-}5)$$

where m is the dimensionality of the vector α and equals one for this study. Although our detector does not extend to inifinity, we can assume that the area lost is neglible if the detector width is much larger than the beamwidth and the beam is kept away from the edges of the detector. Thus, the maximum amplitude of the rate function, Γ, is:

$$\Gamma = [\bar{n}/(t_F-t_0)] \; [(2\pi) \; R \;]^{-1/2} \qquad (4\text{-}6)$$

The mean noise arrival rate is defined as

$$\overline{\lambda}_n(t) = \int_{-L/2}^{L/2} \lambda_n(t,r)dr \qquad (4\text{-}7)$$

where L is the physical length of the detector. For this study $\lambda_n(t,r)$ is assumed to be a constant denoted as λ_n, so that $\overline{\lambda}_n = \lambda_n L$.

4.2.1.2 <u>Truth Models</u>. Adapting SOFE to this application did not affect the propagation equations for the SOFE truth models used to simulate the NPB and target. In this study the truth model propagation equations are linear and given by Equations (2-3) and (3-4). SOFE uses a fifth order Kutta-Merson integrator to solve the deterministic portion of the propagation equations. Then a stochastic term is added to user-specified states at periodic intervals (after the completion of each integration to that time) through separate calls to a Gauss random number generator for each state. Expressed in equation form:

$$\underline{x}_S = \underline{x}_D + \text{GAUSS}(\underline{0}, \underline{Q}_d) \qquad (4\text{-}8)$$

where $\underline{x}_S$ is the stochastic solution, $\underline{x}_D$ is the deterministic solution, and the expression GAUSS($\underline{0}$, $\underline{Q}_d$) is a vector of randomly generated noises of mean zero and a covariance of $\underline{Q}_d$. The discrete-time dynamics driving noise covariance, $\underline{Q}_d$ (the second moment of the equivalent discrete time noise representing the effects of $\underline{G}(t)\underline{w}(t)$ on a state over a user specified propagation interval), is calculated by the user from Equation (2-19) in Section 2.3 for the beam and by Equation (3-21) in Section 3.2.1 for the target.

4.2.1.3 Target Kalman Filter. SOFE propagates the the Kalman filter states in to same manner as the truth states (see previous section). The equation used to propagate the filter covariance $\underline{P}$ in SOFE is assumed to be of the form [24:83]:

$$\dot{\underline{P}}(t) = \underline{F}(t)\underline{P}(t) + \underline{P}(t)\underline{F}^T(t) + \underline{G}(t)\underline{Q}(t)\underline{G}^T(t) \qquad (4\text{-}9)$$

As with the state propagation equations, SOFE uses a fifth-order Kutta-Merson integrator to solve Equation (4-9)

SOFE uses a recursive method to incorporate the Kalman filter measure updates. In this study the measurement model is also linear and given by equation (3-8). SOFE injects the measurement noise into each measurement by the following relationship:

$$z(t_i) = \mathrm{GAUSS}(\underline{Hx}_{TP}(t_i), R) \qquad (4\text{-}10)$$

where R is the variance of the target sensor measurement noise. For this problem the measurement noise is assumed to be mean zero. The standard Kalman filter updates at regular discrete time intervals determined by the user, whereas the Meer filter updates only when it receives a Poisson point process event.

4.2.1.4 <u>Meer Filter</u>. Unlike the method SOFE uses to propagate the (possibly nonlinear) Kalman filter states, the Synder-Fishman filters which make up the Meer filters are propagated using the state transition matrix. This is possible only because of the simple beam model we have chosen (see Chapter 2). The propagation is given by (again in one dimension):

$$\hat{x}(t_{i+1}^-) = \Phi\, \hat{x}(t_i^+) + B_d\, u \qquad (4\text{-}11)$$

where $\Phi = \exp(-\Delta t/\tau_B)$ and $B_d = \tau_B\,(1-\Phi)B$ as given by Equation (3-19), and B is constant for this study. The time interval t is the time since the last Poisson process event occurred. The integrator for the truth model and the Kalman filter model are also provided the time of the next Meer filter update. However, the propagation of the truth states, the Kalman filter states and the filter-computed error covariances is accomplished by integrating the differential equations from the current time forward to some specified time. This time can be the integration step size, the time for the next Snyder-Fishman filter update, or the time for the next Kalman filter update. Thus, the Meer filters propagate to all the predetermined points for control application and also to all the measurement event times. The Meer filter estimates need to be taken at regular intervals for ensemble processing. Therefore, they are recorded at the Kalman filter update times.

As mentioned before, the MMAE Meer filter's measurement update occurs whenever an event is observed. If the event was signal-induced, the true location of event and the measurement is given by:

$$z(t) = \mathrm{GAUSS}(\underline{Hx}_{TB}(t), R) \qquad (4\text{-}12)$$

where $\underline{Hx}_{TB}$ is the true location of the beam centerline on the detector and $\sqrt{R}$ is the beam halfwidth. We have assumed that the detector provides perfect measurements in this study. If the event is noise-induced, Equation (4-12) does not apply and the location of the noise-induced event is simulated by using a uniformly distributed mapping function, uniformly distributed over the length L of the detector.

The updating to the Meer Filter and the MMAE Meer filter follows the method given in Section 2.3. This is followed by the pruning of the Meer filter as described in Section 2.5.2. The last major portion of user modification is that related to the setup and operation of the controller.

4.2.1.5 Controller Simulation. Zicker [32] added the first controller design as discussed in Section 3.3.6. Moose and Jamerson [8; 23] designed the follow-on proportional plus integral controllers and Johnson incorporated the first MMAC design employing proportional plus integral controllers. All of these designs used the basic structure developed by Zicker [32:39] for calling the controller routine. As such, it operates independently of the MMAE Meer filter propagation and update cycle and the Kalman filter propagation and update cycle to provide control feedback at user-specified intervals using the estimates of each filter. Normally the controller update cycle is tied to the Kalman filter cycle. This same calling structure is used in this study so that results can be compared to Johnson's work.

4.2.1.6 Additional Alterations. The only other significant modification to the basic SOFE code is Zicker's alteration of a portion of the Gauss random number generator to increase efficiency [32:42]. The conceptual operation of the modified SOFE program is shown at a macro level in Figure 4-1. SOFE is used to generate the data necessary to determine performance; however it would be impractical, if not impossible, to determine the actual performance from the raw data. Therefore, another software package is used to post-process the data.

4.2.2 Assimilating the Data - SOFEPL.

SOFEPL is a postprocessing routine that can be programmed to perform various statistical functions, such as calculate the RMS error time history, and then plot the results. The data provided by SOFE is statistically reduced and the results are plotted by SOFEPL. First this section will discuss the statistics used to evaluate the performance of the MMAE Mee filters. Although the SOFEPL program provides an array of statistical options, they are limited to the error statistics found between the truth and filter states (for filter performance evaluation) and are inadequate for evaluating a controller design. An alternative approach provided by SOFEPL is for the user to specify the desired error to be evaluated statistically. The final portion of this section discusses the procedure used to generate the controller RMS error time history, where the statistics are based on the error between the true beam and true target positions.

4.2.2.1 MMAE Meer Filter Statistics.

The MMAE Meer filter's state estimation error statistics are used to

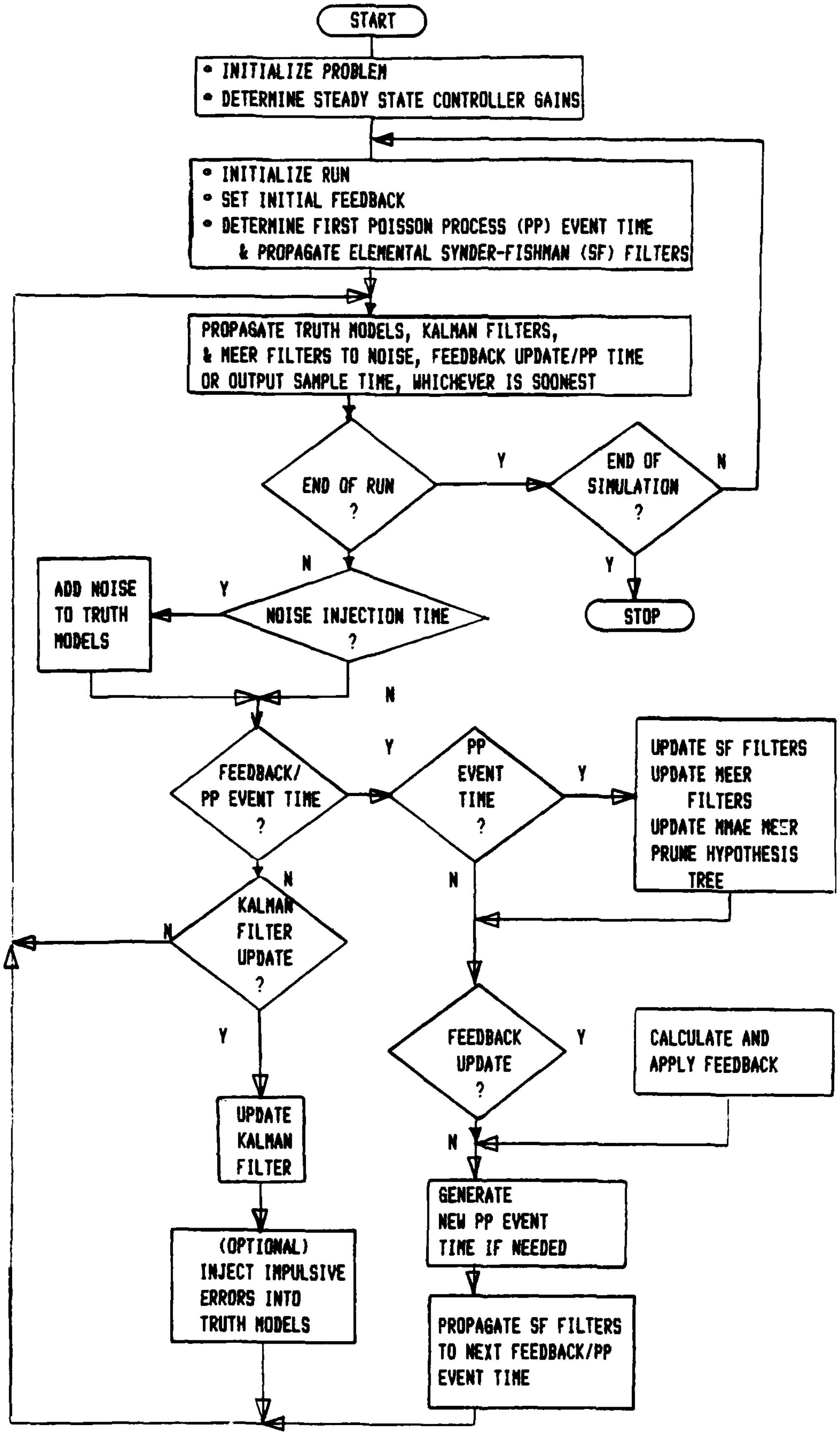

START
• INITIALIZE PROBLEM
• DETERMINE STEADY STATE CONTROLLER GAINS
• INITIALIZE RUN
• SET INITIAL FEEDBACK
• DETERMINE FIRST POISSON PROCESS (PP) EVENT TIME
& PROPAGATE ELEMENTAL SYNDER-FISHMAN (SF) FILTERS
PROPAGATE TRUTH MODELS, KALMAN FILTERS, & MEER FILTERS TO NOISE, FEEDBACK UPDATE/PP TIME OR OUTPUT SAMPLE TIME, WHICHEVER IS SOONEST
END OF RUN ?
Y
END OF SIMULATION ?
N
Y
STOP
N
NOISE INJECTION TIME ?
Y
ADD NOISE TO TRUTH MODELS
N
FEEDBACK/ PP EVENT TIME ?
Y
PP EVENT TIME ?
Y
UPDATE SF FILTERS
UPDATE MEER FILTERS
UPDATE MMAE MEER
PRUNE HYPOTHESIS TREE
N
N
KALMAN FILTER UPDATE ?
N
Y
UPDATE KALMAN FILTER
(OPTIONAL) INJECT IMPULSIVE ERRORS INTO TRUTH MODELS
FEEDBACK UPDATE ?
Y
CALCULATE AND APPLY FEEDBACK
N
GENERATE NEW PP EVENT TIME IF NEEDED
PROPAGATE SF FILTERS TO NEXT FEEDBACK/PP EVENT TIME

Figure 4.1 Macro Level Flowchart of Modified SOFE

evaluate each filter's operation. These statistics are the mean and the standard deviation of the error between the truth and filter states. The derivation of the MMAE Meer filter's beam position state estimate error statistics are as follows. The error between the true beam position and the MMAE Meer filter estimate is

$$e_B(t_i,n) = x_{tB}(t_i,n) - \hat{x}_B(t_i,n) \tag{4-13}$$

where x_{tB} is the true beam position and $\hat{x}_B$ is the MMAE Meer filter position estimate defined in equation (2-15) and n is the Monte Carlo run number. Thus, the mean error time history is:

$$m_e(t_i) = (1/N) \sum_{n=1}^{N} e_B(t_i,n) \tag{4-14}$$

where N is the total number of runs. The variance and the standard deviation of the error are [4:74-77; 6]:

$$v_e(t_i) = [1/(N-1)][\sum_{n=1}^{N} e_B^2(t_i,n)] - [N/(N-1)]\, m_e^2(t_i) \tag{4-15}$$

$$\sigma_e(t_i) = [1 + 1/[4(N-1)]]\, [v_e(t_i)]^{1/2} \tag{4-16}$$

Equation (4-15) calculates the unbiased estimate of the variance for all the runs and Equation (4-16) is an approximation that produces an unbiased (to first order) estimate of the standard deviation for all the runs. A full derivation of the exact equation and the limitations that apply to the approximation can be found in Deming [5; 9:75].

The root-mean-square (RMS) tracking error, $RMS_e(t_i)$, and the time-averaged RMS tracking error, RMS_e, are used to further compress the statistics used to evaluate the performance of the estimater/controller during the different analyses. The time averaged RMS error ($\overline{RMS}_e$) is useful as it provides a single number indicating both the precision and consistency of estimator/controller performance. This single number simplifies the task of comparing sets of statistics. However, the mean error plus and minus one standard deviation time histories are really the most useful information for evaluation of the estimator/controller.

The RMS error time history can be generated by using the equation:

$$RMS_e(t_i) = [m_e^2(t_i) + \sigma_e^2(t_i)]^{1/2} \qquad (4\text{-}17)$$

The time averaged RMS error is calculated by averaging each RMS error, $RMS_e(t_i)$, from an initial time t_0 to a final time t_F by the equation:

$$\overline{RMS}_e = [1/(f+1)] \sum_{i=0}^{f} RMS_e(t_i) \qquad (4\text{-}18)$$

Similar to the MMAE Meer filter error statistics are the controller error statistics discussed next.

4.2.2.2 *Controller Statistics*. The error for the tracker problem is defined as the difference between the true target state and the true beam position state:

$$e_T(t_i,n) = x_{tXP}(t_i,n) - x_{tB}(t_i,n) \qquad (4\text{-}19)$$

where x_{tXP} is true target position generated by the target truth model defined in Section 3.1 and x_{tB} is true beam position generated by the NPB truth model defined in Section 2.1. The errors from each run are averaged and a mean error time history, $m_e(t_i)$, a variance time history, $v_e(t_i)$, a standard deviation of error time history, $\sigma_e(t_i)$, the RMS error time history, and the time averaged RMS error are calculated using Equations (4-13) through (4-19) as with the beam error. Equation (4-19) is also used to evaluate the filter computed tracking error by replacing the actual target and beam states with the estimated states from the MMAE Meer and Kalman filters. The actual rms error that occurs is dependent on the parameter values used in the truth models, filters and controller. The next section will present the parameters used in this study.

4.3 Filter, Controller, and Simulation Parameters

The particle beam estimator/controller problem is modeled as a stochastic process and the performance can be evaluated by observing the statistical behavior of the estimation and control tracking error processes. We are interested in both the sensitivity and the robustness of our estimator or controller design. In evaluating the controller's sensitivity and robustness, we would like to analyze tracking error statistics for all physically reasonable variations of the true beam and true target parameters. However, the time constraints of doing this study require that only a small variation of selected beam parameters and target parameters be considered for sensitivity analysis, and similarly only a small variation in selected filter parameters be considered in robustness analysis. First this section reviews the beam, target and simulation parameters. Then it presents the nominal values for each parameter in this study and the sizes of variations to be considered.

4.3.1 <u>The Beam and Target Parameters</u>. This section reviews the MMAE Meer beam estimator and target filter parameters used in this study. The MMAE Meer filter is designed on the basis of six parameters. They are as follows:

τ_B is the beam time constant, defined in Equation (2-3b). This parameter is different for each elemental Meer filter in the bank structure of the MMAE algorithm

g is the square root of the beam dynamics driving noise strength, g= (G Q)   as defined in Equation (2-3b)

R is the beam dispersion measured as the analog of variance of the Gaussian-shaped beam signal rate function at the detector surface; See Section (2.2), Equation 2.7

λ_n is the noise arrival rate per length of the detector array; see Equation (2-8)

$\bar{n}$ is the expected number of signal-induced events during a simulation run; see Equation (2-10)

D is the elemental Meer filter memory depth; see Section 2.5

L is the length of the one-dimensional detector array; as discussed in Section 2.2.1, the appropriate minimum is twelve times the beamwidth

SNR is the signal-to-noise ratio as defined in Equation (2-10):

$$\begin{aligned} SNR &= [\bar{n}/(t_F - t_0)]/[\lambda_n L] \\ &= \Gamma \sqrt{(2\pi R)} / [\lambda_n L] \end{aligned} \qquad (2\text{-}10)$$

Since $\bar{n}$, the number of signal-induced events expected during one simulation run, $(t_F - t_0)$, the duration of one simulation run, λ_n, the noise arrival rate per length of detector, and L, the length of the detector, are all independent variables, then SNR is a dependent variable. Therefore, it should be determined from the other parameters. This is in contrast to previous thesis efforts where the noise rate was made to be a dependent variable using Equation (2-10).

The three parameters associated with the 3rd order Gauss-Markov position process target truth model and target Kalman filter are as follows:

τ_T is the target time constant defined in Equation (3-3)

Q_T is the target dynamics driving noise strength defined in

Equation (3-21)

R_T is the target position measurement noise variance defined in Equation (3-8b)

A complete analysis would include the effects of varying all the parameters throughout their physically realizable ranges. Time constraints prevent such a through analysis and only a few of the parameters are considered in this study. These parameters are presented next.

4.3.2 Nominal Parameters. Actual estimator and/or controller performance is related to the parameters used to design it and the conditions under which it operates. It is therefore very important to know the design parameters of the filters and controllers as well as the operating parameters found in the truth model. The following parameters are provided as a benchmark for comparison with later studies. The nominal values for the beam and tracker filter parameters used for the analyses are as follows:

Nominal Beam Parameters

τ_B = 20 sec   D = 1

g = 0.2 cm/sec$^{1/2}$ R = 0.5 cm^2

$\bar{n}$ = 100 signal events

Since Johnson's sensitivity analysis (the filter-assumed parameter matches the true parameter in sensitivity analysis) found less than 1% variation in performance for order of magnitude changes in the beam time constant, the nominal value for this parameter will remain at 20 seconds.

The other beam parameters also remain the same as those used by Johnson so that results will be directly comparable.

Nominal Target Parameters

τ_T = 10 sec R = 0.5 cm^2

Q #1 = 0.01 $\mathrm{cm}^2/\mathrm{sec}^5$

Q #2 = 0.07 $\mathrm{cm}^2/\mathrm{sec}^5$

Q #3 = 1.00 $\mathrm{cm}^2/\mathrm{sec}^5$

The three values of Q_T correspond to the three values chosen by Johnson [9:69] for adaptively modelling the target noise strength. Johnson describes these three values as modelling the trajectories from the straight and level flight, to the maximum manned-vehicle g-limit of 10 g's.

The Simulation Parameters

Zicker studied the number of Monte Carlo runs necessary for the sample statistics to converge to a consistent value [32]. Based on Zicker's work, the number of Monte Carlo runs, N, is set at 200. This large number is necessary due to the sparse number of photon events.

Because the fastest transients at the nominal condition have a time constant of ten seconds (rather benign dynamics), the Kalman filter and the feedback control sample period will be set to one second. This easily satisfies the Shannon sampling theorem which states that the sampling rate should be at least twice the highest signal frequency content of interest [33:87-88]. One second also provides the extra margin suggested by the engineering rule of thumb, of sampling five to ten times faster than the highest frequency used [33:497].

As previously noted, the detector length is constrained by the beamwidth. However, the choice of detector length is further constrained by the scaling chosen and computer limitations for exponent arguments. The value of 20 cm for L allows adequate size for beam dispersions from .3 cm^2 to .7 cm^2, while maintaining exponent arguments within computer defined limits. This L is adequate for this study and it is desirable to maintain the same scale with past studies, however future studies may wish to apply scaling factors to the problem to avoid limitations on exponent argument limits.

To provide a λ_n that is consistent with the values used in the past (λ_n corresponding to SNR = 20 and L = 10), when the mean of signal rate was 1 event/sec, λ_n is set to 0.005 events/sec/cm. Since the detector length is now 20 cm, the equivalent SNR is now 10. Having a constant noise rate per unit length, rather than one dependent on the SNR, L, expected number of signal events, and duration of the run, will provide more realistic results for changes in signal event rate, and detector length.

4.3.3 <u>Analysis Parameters</u>. Since Johnson experienced significant degradation in controller tracking performance with only 2% variation in true beam time constant from the filter assumed value [9:118], this is the focal point for robustness analysis of the controller. The 2% variation is also the starting point for the MMAE estimator performance evaluation. Larger variations of the true beam time constant from the nominal, filter-assumed value may be studied if beam estimator performance or controller tracking performance at 2% variation warrants further study. Johnson found no sensitivity problems with variations of the beam time constant nor significant sensitivity or robustness problems with other beam parameters [9]. The inclusion of the MMAE structure within the controller may result in a change to the sensitivity or robustness of the closed loop system. Therefore, sensitivity and robustness analyses are performed for variations in the beam dynamics driving noise strength and beam dispersion. The variation considered will be one order of magnitude above and below, with a nominal value of 0.2 cm^2/sec for the beam dynamics driving noise strength and a nominal value of 0.5 cm^2 for the beam dispersion. This will allow comparisons to be made with Johnson's results [9]. The next chapter will describe the results obtained for simulation runs based on the nominal parameters given in this section.

4.4 Summary

The purpose of this chapter has been to explain the tools and methods used to evaluate estimator performance of the MMAE Meer filter and to evaluate the controller performance of the MMAC based on weighting matrices generated through implicit model following techniques. The Monte Carlo simulation provides the most viable method of evaluating the estimator's or controller's performance. SOFE and SOFEPL provide the basic method of computing the filter state estimation error statistics and controller tracking error statistics, which allow us to evaluate controller designs. Four of the performance analyses: controller performance with MMAE Meer filter, controller performance with reduced depth MMAE Meer filter, controller performance with the expanded bank MMAE Meer filter, and the time-varying parameter analyses, determine the capabiltiies of the MMAE Meer filter to handle off-nominal beam time constants in a closed-loop system. Three of the performance analyses: MMAC performance with weighting matrices generated from implicit model following techniques, MMAC sensitivity investigation, and MMAC robustness analysis, evaluate the capabilty of weighting matrices based on implicit model following techniques to provide additional robustness to the closed-loop system incorporating the MMAE Meer filter while maintaining acceptable performance at design conditions. The results of the seven analyses are in the next chapter.

V. RESULTS AND ANALYSIS

The results of the eight Monte Carlo analyses discussed in Chapter 4 are presented in this chapter. The first five analyses are on the MMAE Meer filter operating in open loop as a beam estimator. The last three analyses are on the MMAE Meer filter operating in closed loop. In this mode, the MMAE Meer filter provides state estimates to a controller which uses the beam estimate, combined with target state estimates, to generate a control input to point the beam at the target. The controller used is the three-element MMAC developed by Johnson [9] with constant gains based on the nominal beam time constant.

The first analysis demonstrates the MMAE Meer filter can provide precise estimates of the beam location when the beam time constant is uncertain. The second analysis shows that the same MMAE Meer Filter does not provide a precise adaptive estimate of the beam time constant when the beam is not in a closed loop. The third analysis indicates use of an elemental Meer filter depth of one is adequate for state estimation purposes. The fourth MMAE Meer filter analysis shows that the filter can perform well in the presence of a linearly changing or a sinusoidally varying true beam time constant. The final MMAE Meer filter analysis indicates that a simple bank of 3 Meer filters performs well for a broad range beam time constants.

Operation of the MMAE Meer filter in a closed loop tracker results in a change of performance for the MMAE Meer filter from that seen in the open loop analysis. To ensure that signal events occur on the array while the beam is tracking the target, two methods of keeping the beam over the detector were implemented and analyzed. The first is, conceptually, the rotation of the NPB toward the target. The second is, conceptually, the sweeping of the detector underneath the beam. Subsequent analyses employ the second approach. The performance of the MMAE Meer filter is then reanalyzed for closed loop operation. Along with this analysis, tracking performance baselines are provided against which subsequent controller designs may be compared.

Time constraints prevented the analyzing the performance of the nine-element MMAC based on implicit model following weighting matrices. However, performance of this MMAC is presented, using the state and control weightings of:

$$X_{11} = 100, \quad X_{55} = 10, \quad U = 1$$

where these are the quadratic cost weighting matrices on the target tracking error, the pseudointegral state and control, respectively. This allows a comparison of this MMAC with the baselines and past controller designs.

5.1 MMAE Meer Filter State Estimation Performance

The first analysis indicates the MMAE Meer filter can provide precise estimates of the beam location when the beam time constant is uncertain. The study uses an MMAE Meer filter composed of three elemental Meer filters. The beam time constants for Meer filters 1, 2, and 3 are 20.0 20.8 and 19.2 respectively, representing a nominal beam time constant plus and minus four percent. Each Meer filter is provided the true value of the beam dispersion and the dynamics driving noise strength. Since all the Meer filter and MMAE Meer filter variables have to be sampled at regular intervals for ensemble averaging of the Monte Carlo run, it is necessary to propagate the state estimates of each individual Meer filter and the MMAE MMAE state estimate. Thus, it is necessary to select a beam time constant to propagate the MMAE Meer filter state estimate to each sample time. The choices considered were the adaptive beam time constant of the MMAE filter and the nominal time constants of the three filters. An initial run, using the adaptive beam time constant to propagate the adaptive state estimate to the sample times, was made with true beam time constant of 20.4 seconds. This run indicated that the estimate of the beam time constant was poor. Therefore, the nominal beam time constant is used to propagate the state estimate in subsequent runs of the MMAE Meer filter in open loop. In addition, the Meer filter initial state variances were adjusted from the past value of .1 to .4. The latter value is the mean square value for centroid oscillations, representing the steady state motion of the beam centroid, as derived from Equation (2-16):

$$P_{SS} = 1/2\ \tau_B\ g^2 \qquad (5\text{-}1)$$

The rms errors for various values of the true beam time constant are in Table 5.1. As the table indicates, the RMS error for lower beam time constants is smaller. This trend is due to the lower steady state oscillation of the beam as given by Equation (5-1). Figure 5.1 is the MMAE Meer mean error plus and minus one standard deviation time history for true beam time constant of 20.4 seconds. However, Figure 5.1 is also representative of the elemental Meer filter performance, regardless of the true value of the beam time constant. The central jagged line is the mean error history of the MMAE estimate from the true beam position. The smooth dashed lines are the envelope of plus and minus the filter's predicted error standard deviation or square root of the filter-computed variance. The outer jagged lines are the mean error ± one standard deviation envelope. What is

Table 5.1 MMAE Meer State Estimate Performance

$R = .5\ cm^2 \quad g = .2\ cm/sec^{1/2} \quad$ Depth = 3

True beam time constant (sec)	RMS error (cm)
20.4	.377023
20.0	.376193
19.6	.374878

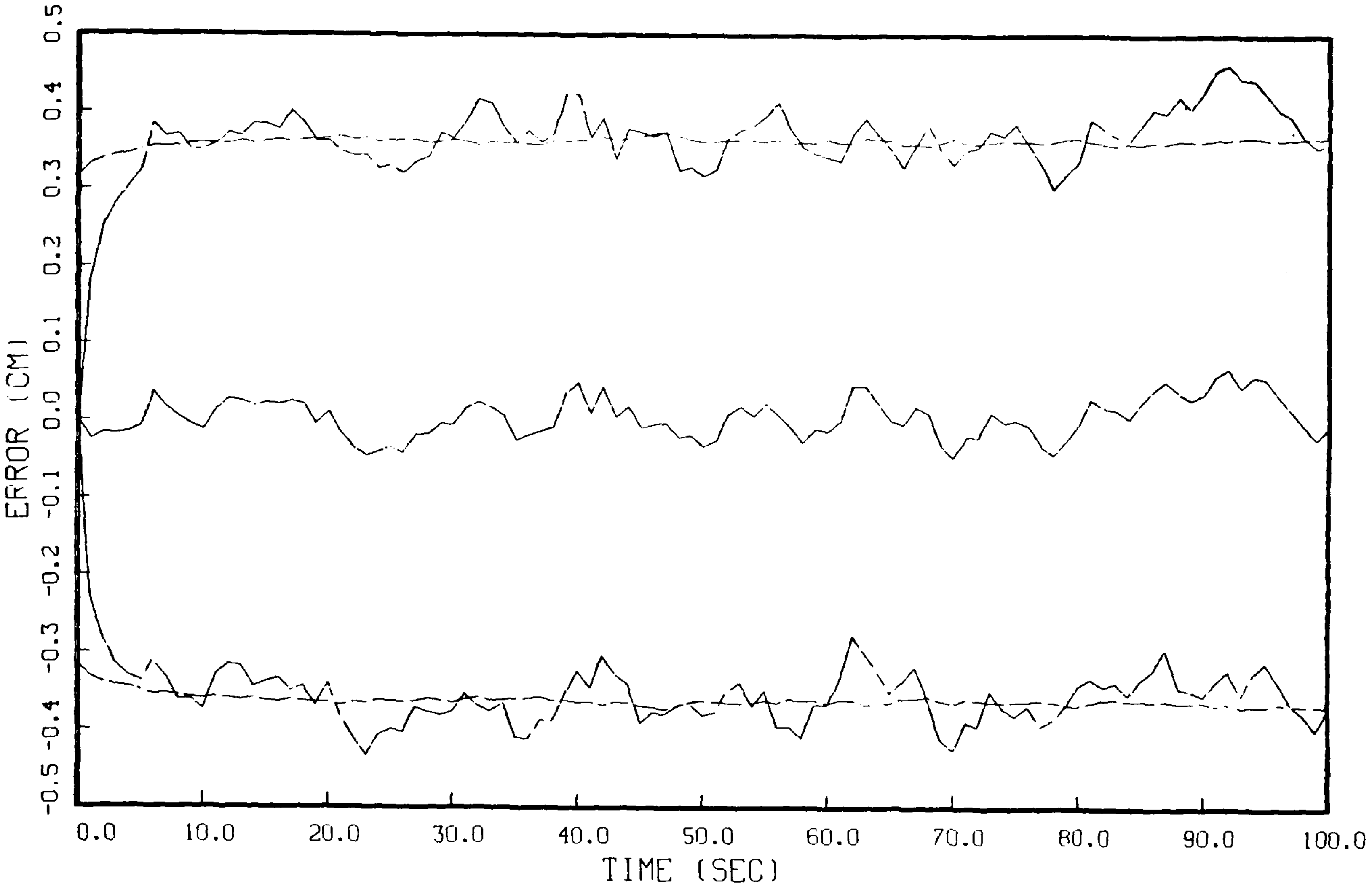


Figure 5.1 MMAE Meer State Estimate Mean Error and Standard Deviation; True Tau = 20.4

significant in the plot is that the mean error is near zero and the filter predicted-error standard deviation remains near the true standard deviation of the error. The performance of the individual Meer filters is given in Table 5.2. As noted, there is very little variation in individual Meer filter performance for changes in the beam time constant. Thus, the Meer filter has little difficultly in estimating the beam location with a mismodelled beam time constant for an uncontrolled beam. As previously discussed, the RMS errors drop as the true beam time constant is reduced due the lower steady state oscillations of the beam. Also note that the Meer filter with the highest filter-assumed beam time constant has the lowest RMS error of the three Meer filters. Again this is related to the steady state error variance. The larger filter-assumed beam time constant results in a larger filter computed variance. This larger variance results in a slightly larger gain placed on incoming measurements. This results in a better estimate of the beam location and consequently lower RMS error. Table 5.3 is presented to show the similarity in performance between the filters. No other conclusion should be drawn from Table 5.3 as residual performance for the Meer filter involves both good tracking of the beam (resulting in lower residuals) and good rejection of noise events (resulting in high residuals).

Figure 5.2 shows the residual mean history of an elemental Meer filter. As before, this figure is

Table 5.2 Individual Meer State Estimate Performance

R = .5 cm^2 g = .2 cm/sec$^{1/2}$ Depth=3

True beam time constant (sec)	RMS error (cm)		
	Meer#1 τ_B=20.0	Meer#2 τ_B=20.8	Meer#3 τ_B=19.2
20.4	.377067	.376949	.377254
20.0	.376225	.376138	.376378
19.6	.374901	.374893	.374969

Table 5.3 Meer Filter Pseudo-Residual Performance

R = .5 cm^2 g = .2 cm/sec$^{1/2}$ Depth = 3

True beam time constant (sec)	RMS residual (cm)			
	Adaptive	Meer#1 τ_B=20.0	Meer#2 τ_B=20.8	Meer#3 τ_B=19.2
20.4	.129923	.129914	.129956	.129869
20.0	.129907	.129900	.129942	.129854
19.6	.129921	.129913	.129955	.129869

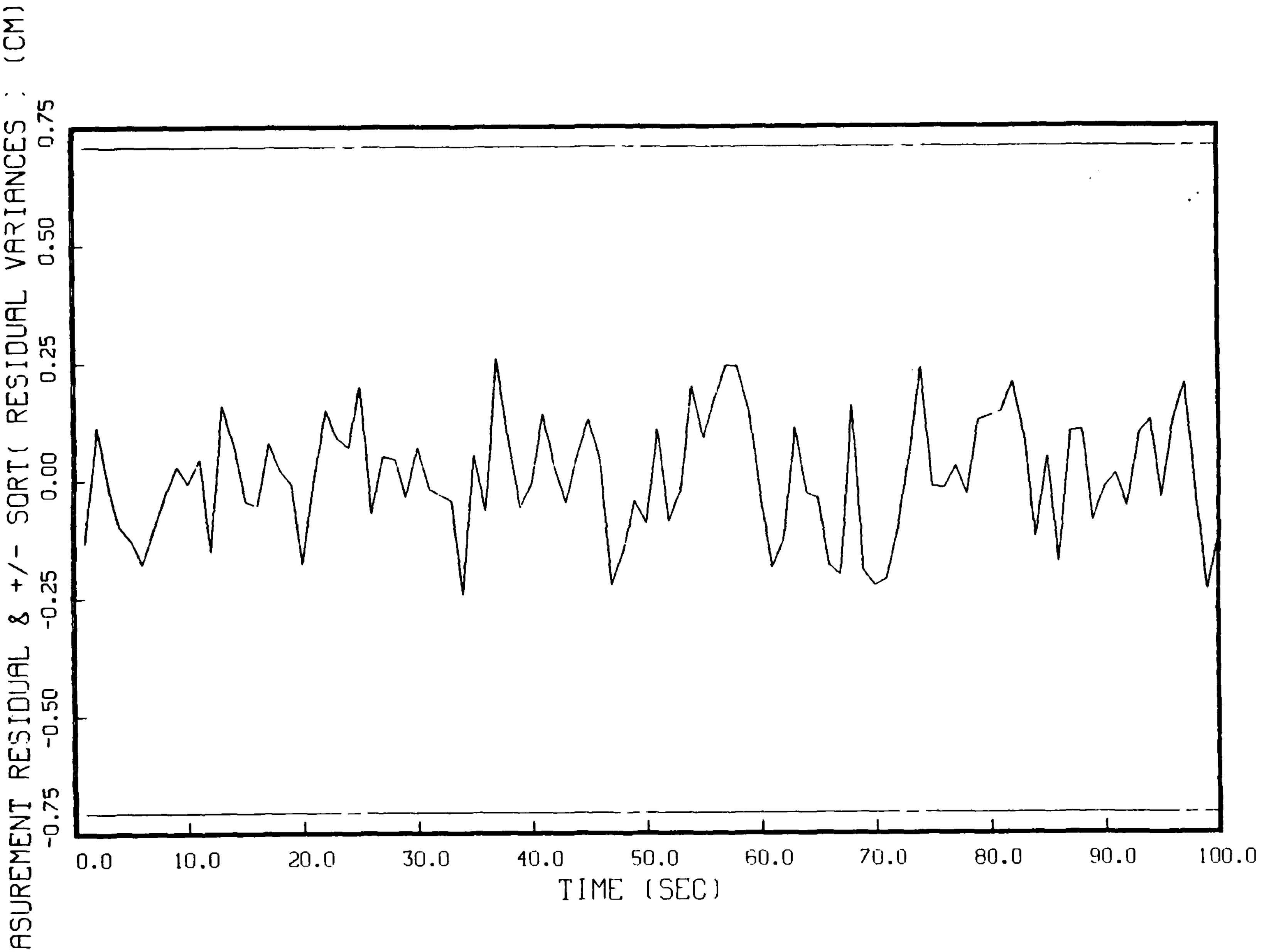


Figure 5.2 Meer Filter #1 Mean Residual

representative of the residual history of any of the Meer filters. Of importance here is that the mean of the residuals remains within the filter computed ±1σ bounds, shown on the plot as dashed lines. Of equal importance is the fact that the results vary so little with changes in beam time constant. It should be noted that these are pseudo-residuals. As with the state estimates, it is necessary to collect the residuals at a uniform sampling interval to allow ensemble averaging. The sampling interval is one second to match the average number of expected signal events. However, Meer filter measurements occur at random intervals. If two events occur during a sample period, then only the residual from the second event is processed. If no events occur during the sample period, then the previous residual is processed again. Thus, some residuals are not collected for processing while a few may be processed more than once. Therefore, these pseudo-residuals are only used to show gross trends in the data. As the performance is remarkably good, no additional Meer filters were incorporated into the MMAE structure. The next results are on the adaptive beam time constant.

5.2 MMAE Meer Filter Parameter Estimation Performance

The second analysis indicates the MMAE Meer filter does not provide precise estimates of the beam time constant when the beam is operated in open loop. This analysis was conducted in conjunction with the first analysis. The beam time constants for each Meer filters are again, 20.0, 20.8 and 19.2 seconds. The average values, over the 100 second interval of the runs, for the mean value of the adaptive beam time constant and the average value of the filter computed variance of the adaptive estimate are in Table 5.4. Of importance here is that once again there is little variation in the performance for changes in the beam time constant. The difference in the variance for the run with true beam time constant of 20.4 seconds is due to an error in setting the initial value of the variance to 0 for that run. As one might expect from the results of the first study, the parameter estimation is very poor. This is due to the small difference in the performance between the Meer filters. Under these conditions one would expect the adaptive parameter estimate will tend toward the filter with the smallest $|\underline{A}_k|$, where $\underline{A}_k$ is defined in Section 2.4 Equation (2-33) [17:133]. Therefore, one would expect the

Table 5.4 MMAE Meer Parameter Estimate Performance

$R = .5\ cm^2$ $g = .2\ cm/sec^{1/2}$ Depth = 3

True beam time constant (sec)	Adaptive beam time constant (sec)	Filter computed Variance
20.4	20.1022	.408476
20.0	20.1021	.432341
19.6	20.1019	.432323

estimate would tend toward the parameter estimate associated with the filter with the smallest beam time constant as that would result in the lowest steady state variance as given by Equation (5-1). However, for the Meer filter, this is offset by the improved residual performance of the filter with the largest beam time constant. The effect of this slight improvement in residual performance is higher probabilities for the filter with the largest $|\underline{A}_k|$. Figure 5.3 depicts the time history of the adaptive beam time constant. Figures 5.4 through 5.6 show the individual filter probabilities for a beam time constant of 20.8 seconds. Again these figures are representive of the performance with any selection of true beam time constant. As with the pseudo-residuals, the adaptive beam time constant and the probabilities are calculated at measurement times, but are recorded at one second intervals for ensemble averaging of the Monte Carlo data. Figure 5.3 might at first appear to indicate that the beam time constant is slowly approaching the correct value. Actually, it is slowly moving toward the value of 20.8 seconds of Meer filter #2, as was explained. To demonstrate this, a Monte Carlo run extended to a final time of 300 seconds is displayed in Figure 5.7. The adaptive beam time constant is prevented from actually reaching 20.8 seconds due to a lower bound on probability limit on each estimator of .1. Thus, the highest adaptive estimate possible with a lower probability bound of .1 is only 20.56. The 0.1 bound was

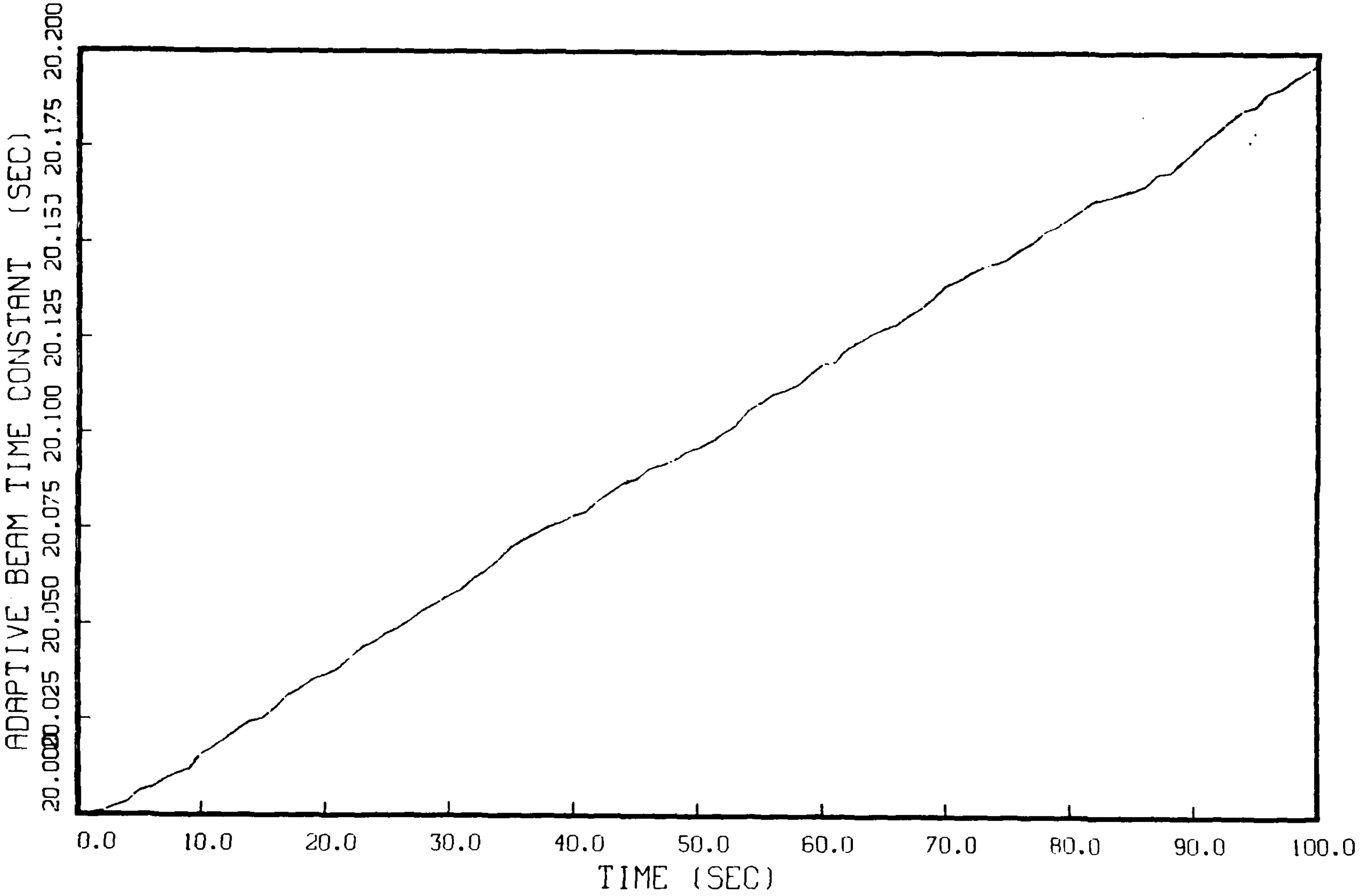


Figure 5.3 Adaptive Beam Time Constant
True Tau = 20.4

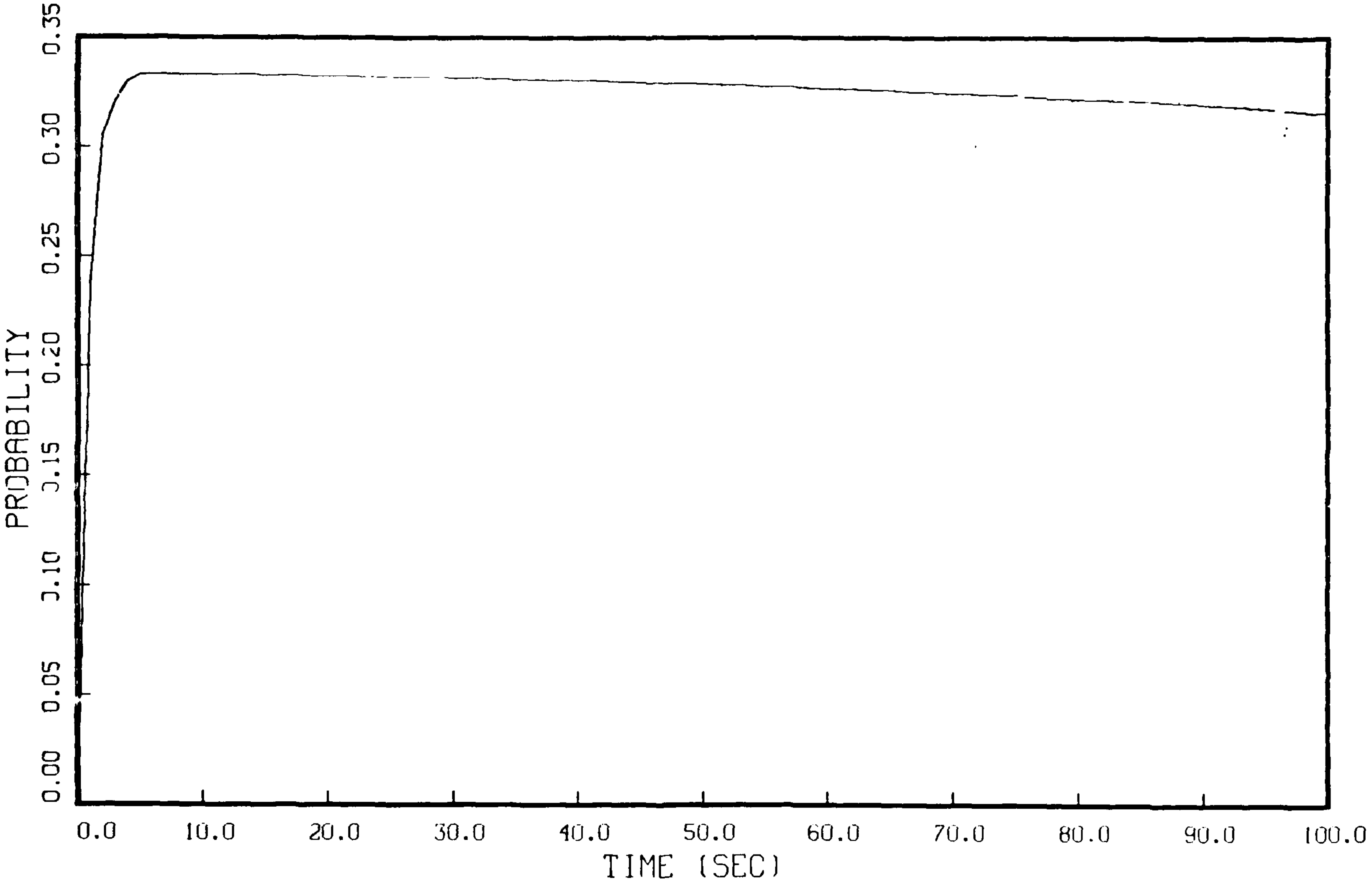


Figure 5.4 Probability History of Meer Filter #1
Filter Tau = 20.0

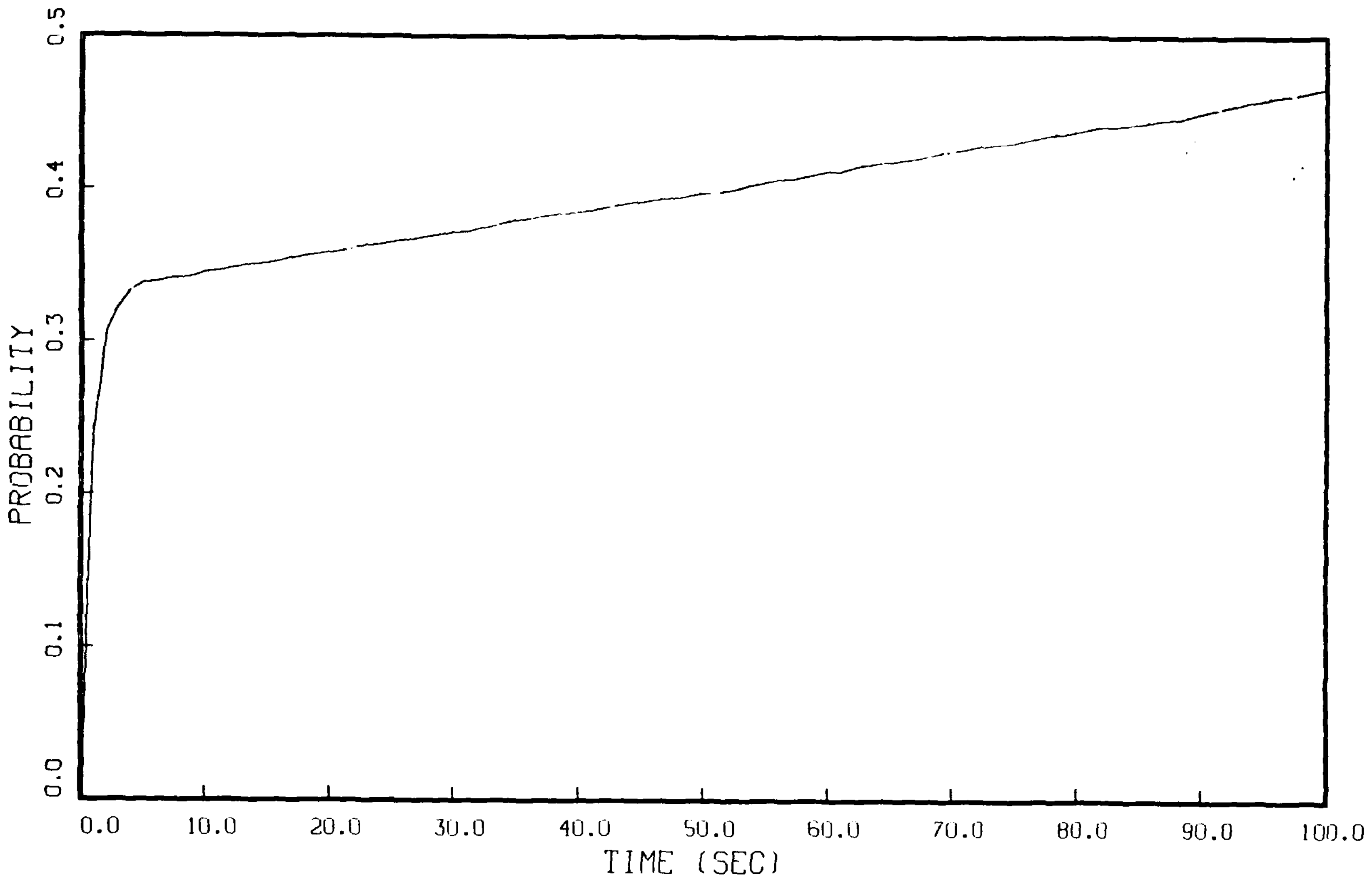


Figure 5.5 Probability History of Meer Filter #2
Filter Tau = 20.8

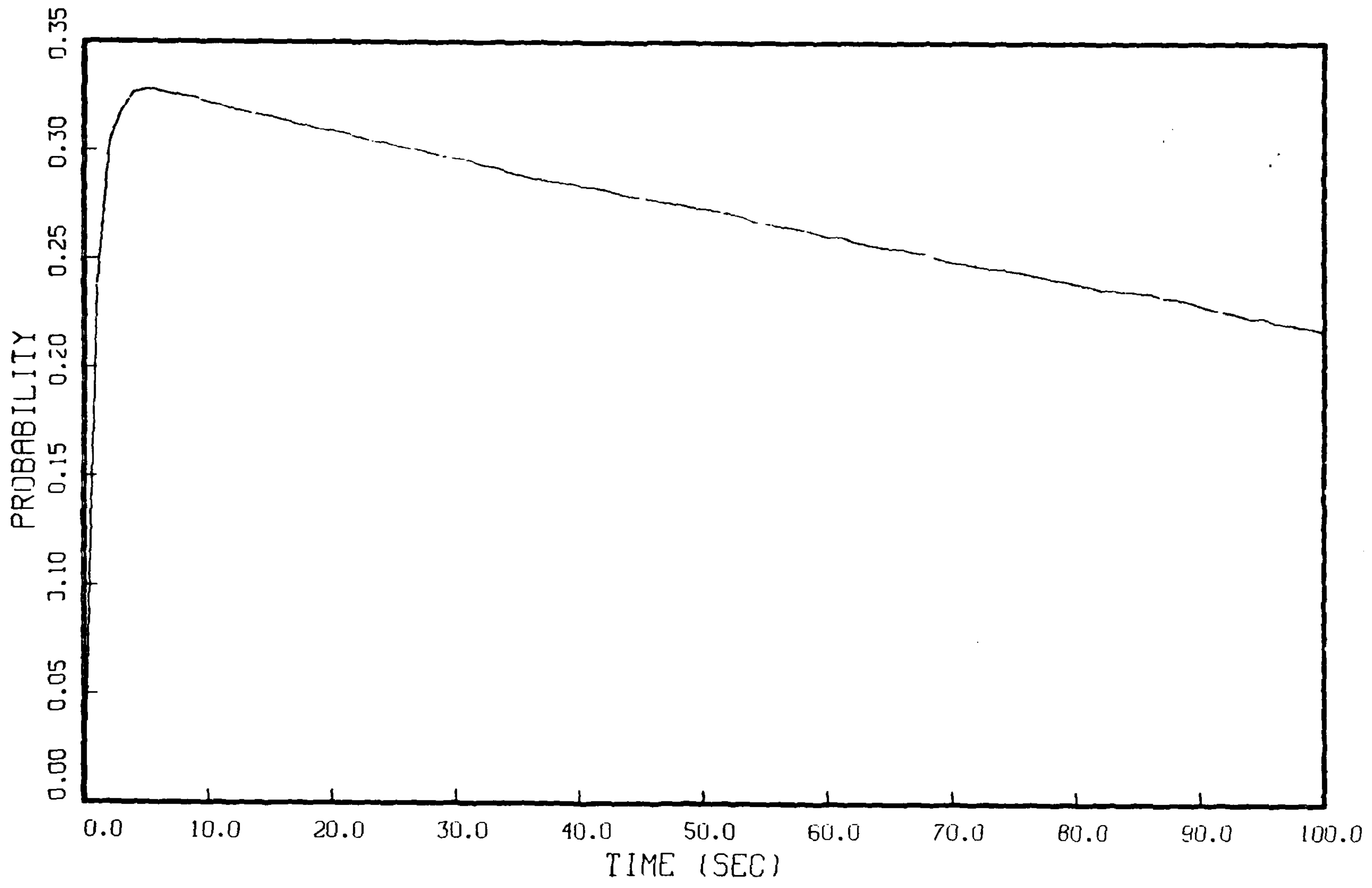


Figure 5.6 Probability History of Meer Filter #3
Filter Tau = 19.2

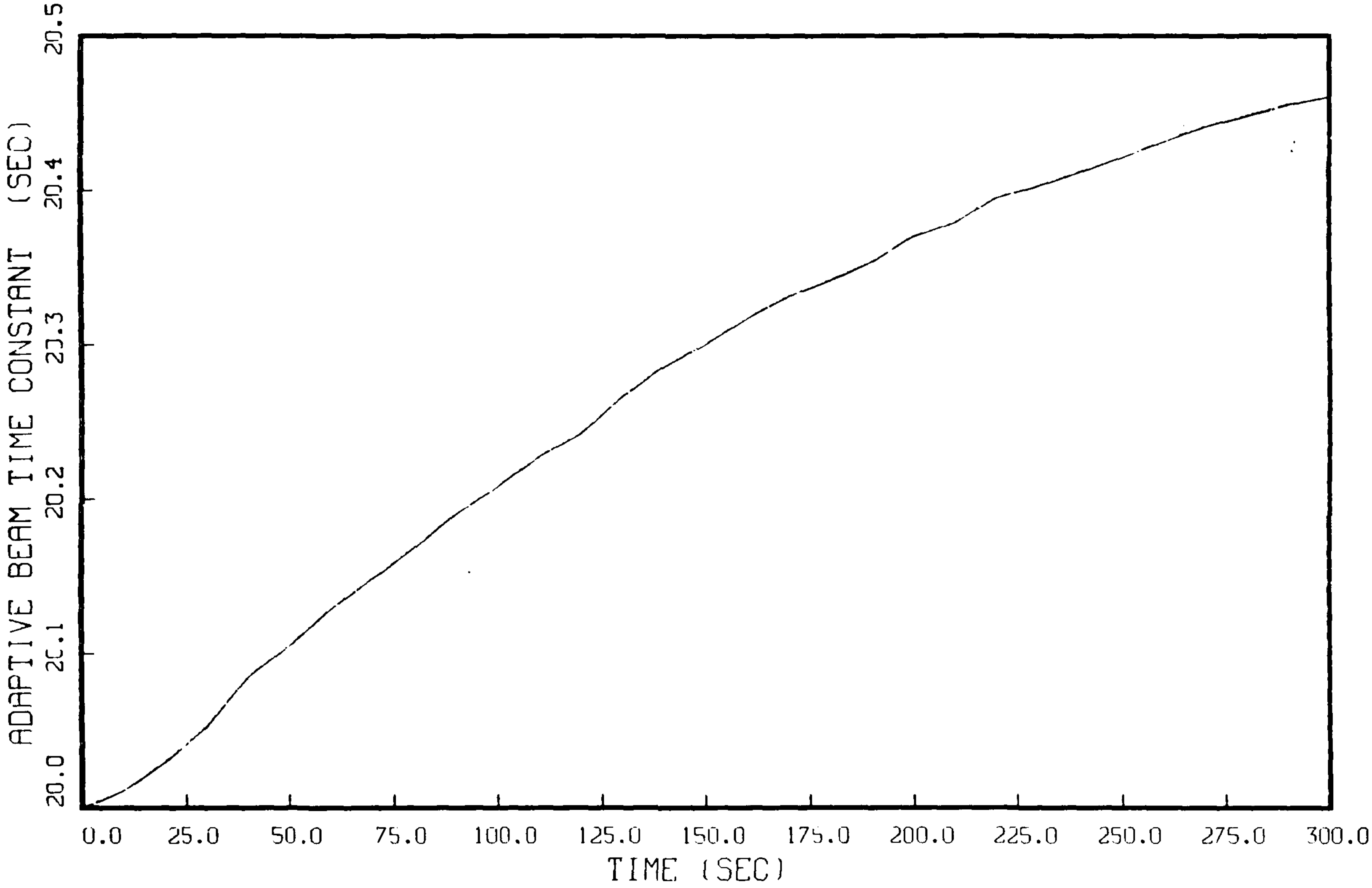


Figure 5.7 Adaptive Beam Time Constant Extended Run

selected to match the lower bound set by Johnson [9] for the Kalman filter probabilities in the MMAC.

The intent of the .1 probability bound is to prevent delays in switching between a controller using a Kalman filter based on a high Q value for a highly dynamic target and a controller based on a Kalman filter with a low Q value for a benign target. However, a bound of .1 is generally considered to be excessive even for this purpose. Later runs use a revised Meer filter probability bound of 0.0005 as a more appropriate value for rather benign beam dynamics. This value allows the adaptive value to approach the upper and lower limits, without allowing a filter's probability to drop to zero. If the filter's probability ever went to zero, then it would no longer have an impact on the adaptive state or adaptive parameter estimate, regardless of the actual beam time constant. A lower bound that is too small might delay the response to a dynamic beam time constant. For the beam in open loop, the lower probability bound has little impact for the short term case, as there is little variation in the Meer filter probabilities as indicated by Figures 5.4, 5.5 and 5.6. Later runs with the new lower bound verified this conclusion.

5.3 Simplified MMAE Meer Filter State Estimation Performance

The third analysis indicates use of an elemental Meer filter depth of one is adequate for state estimation purposes. In this study, special coding is used to implement the simplified update routine described in Section 2.5.2.2 Equation (2-46). In addition, the Meer filter propagating routine is streamlined for the Depth = 1 case. As indicated in Table 5.5, the MMAE Meer state estimation performance showed little difference at a depth = 1 compared to depth = 3. This is not surprising when the effect of the filter depth on the Merge method of Section 2.5.2.2 is considered. The result of increasing the filter depth is to extend the amount of time that a measurement affects the Meer filter estimate (and the MMAE Meer estimate). As such, noise measurements continue to have an effect on the estimate until eliminated from the filter. The purpose of the Merge method is to retain a greater portion of the measurement history in collapsing the filter hypothesis tree than can be achieved by acting on a decision to discard half of the measurement hypotheses as in the "Best Half" method of Section 2.5.2.1. So a measurement retains an indirect

Table 5.5 State Estimate Performance and Computional Loading For Depths of 1 and 3

$R = .5\ cm^2$ $\quad g = .2\ cm/sec^{1/2}$

True beam time constant (sec)	Depth = 1		Depth = 3	
	RMS error (cm)	CPU time (sec)	RMS error (cm)	CPU time (sec)
20.4	.376520	955.099	.377031	966.989
20.0	.375706	950.056	.376193	964.350
19.6	.374878	966.491	.375341	966.341

impact on the state estimation after leaving the filter. Thus, a depth of one still retains much of the measurement history while also minimizing the impact of noise events.

As shown in the table, there is a great deal of time variation between runs. This variation is due to computer loading at the time of the run. Most runs were conducted during evening hours when computer loading is light. A notable exception is the run with true beam time constant of 19.6 and Depth of 1, that was run mid-morning. Comparison of runs made at similar times, indicate there may be some slight reduction in computer time when using a depth of 1. Two reasons account for such a small change in computation time for a change in depth. First, the code was altered to provide greater sensitivity to all depths. As an example, the probability calculation associated with Equation (2-40) was performed for 256 elemental Snyder-Fishman filters regardless of depth in the software of past studies. The use of 3 Meer filters would have tripled this number, however, the code for this study was altered so only 24 probability calculations were required at a depth of 3 for all 3 Meer filters. This is only 8 times the number of probability calculations necessary for 3 Meer filters at a depth of 3. Second, as indicated in the SOFE manual [24:83], the routines involved in propagating the truth and Kalman filter states via the Kutta-Merson integration routine are called most often. This is a substantial portion of the computer load. The elemental Snyder-Fishman filter propagation is accomplished separately, using a one dimensional state transition matrix. Changes in the Meer filter depth will not have a large impact unless the signal rate is high. Even so, the computer loading will still be dominated by the integration routine as it propagates the truth and filter states to the Poisson event times, as well as, noise injection and Kalman update periods. To calculate the loading more specifically due to the Meer filter operations one could count the number of arithmetic operations per Meer cycle for changes in depth. However, as the Meer filter is one dimensional (thus one state and one variance), the number of arithmetic operations will increase by 2 raised to the Depth (D) power. Thus, the Meer filter computation load for a depth of 3 as compared to a depth of 1 is 8 times greater. If the Kalman filter were also propagated by a state transition matrix, the same approach could be used to establish the relative loading between the Kalman and Meer filters. However, that analysis was not performed in this study. Based on the results in Table 5.5, the remaining analyses were conducted at Depth = 1, with the Simplified MMAE Meer filter code.

5.4 MMAE Meer Performance With Variable Beam Time Constant

This study consisted of altering the beam time constant of the truth model without informing the MMAE Meer filter. As before the beam time constants of filters 1, 2, and 3 were 20.0, 20.8, and 19.2 seconds respectively. The true beam time constant was varied from an initial value of 19.2 seconds by adding a factor of 0.016t to the beam time constant in the propagation routine. These values were chosen to span the range of beam time constants. The result is a ramping beam time constant. This was then repeated for a descending ramp, with an initial true beam time constant of 20.8 and slope of -0.016t. Table 5.6 has the state estimation performance under these conditions. Figure 5.8 depicts the mean adaptive beam time constant for the case of the true beam time constant descending linearly from a value of 20.8 to 19.2. As indicated in Figure 5.8, the adaptive beam time constant is also ramping. However, the positive slope matches that seen in Figure 5.3, rather than the negative input slope of -0.016t. Again, this figure is characteristic of the results obtained for the beam time constant with a positive ramp and for later results with a

Table 5.6 <u>State Estimate Performance With Variable Beam Time Constant</u>

$R = .5\ \text{cm}^2$ $g = .2\ \text{cm/sec}^{1/2}$ Depth = 1

True beam time constant (sec)	RMS error (cm)
19.2+.016t	.375747
20.8-.016t	.375646
20.0+.8sin(omega t)	.375633

omega = 2π/100

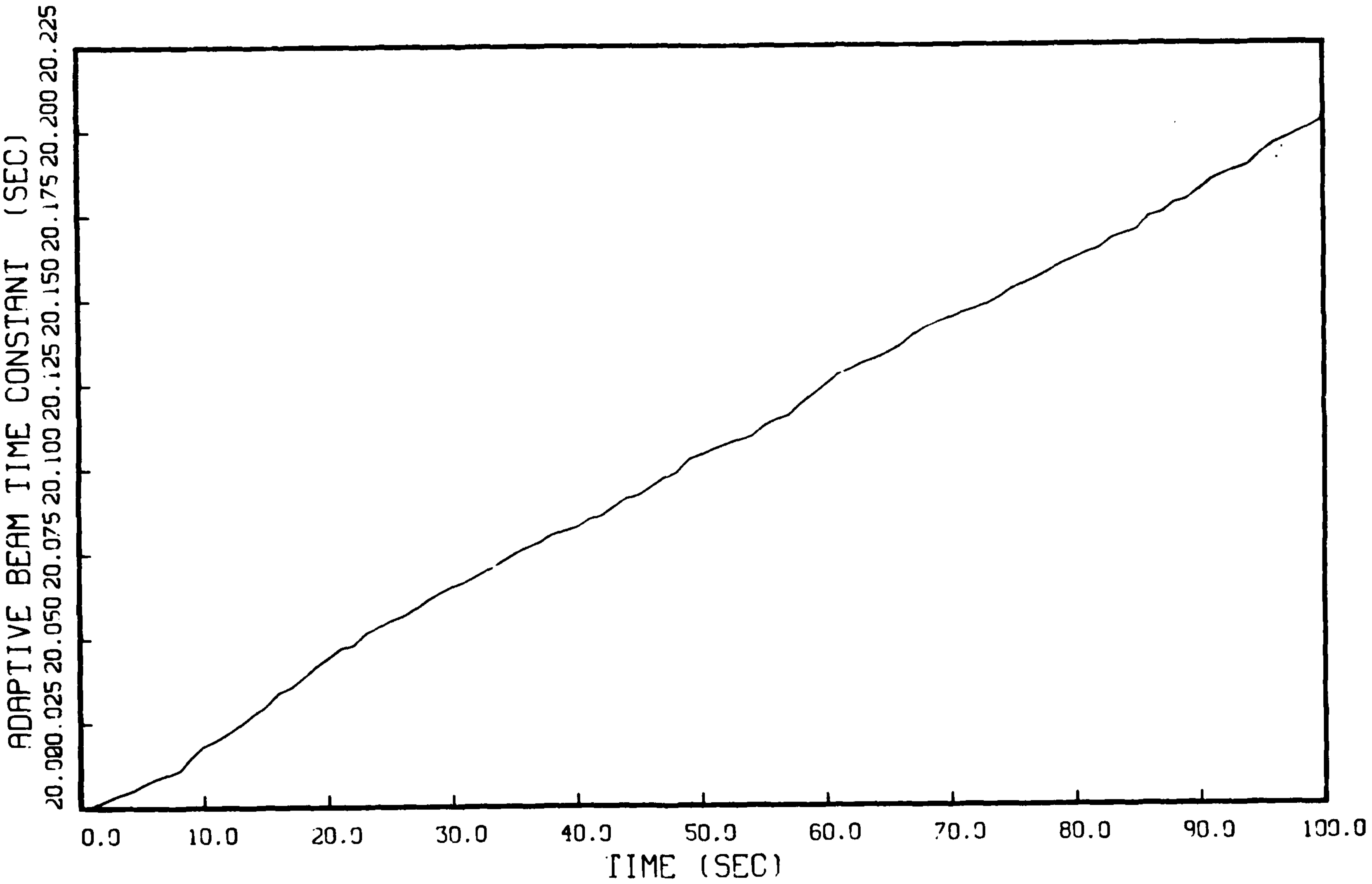


Figure 5.8 Adaptive Beam Time Constant; Downramping Tau
True Tau = 20.8 - .016t

sinusoidally varying beam time constant.

Next, the beam time constant was varied in a sinusoidal manner by adding a factor of 0.8 sin(omega t). The initial beam time constant was again 20.0 seconds. A magnitude of 0.8 was chosen to match the range of beam time constants of the filters. The argument of the sin function, omega, was chosen to be $2\pi/100$ to provide a complete cycle in 100 seconds. The results are also in Table 5.6 and again are similar to previous results.

5.5 MMAE Meer Filter Range With Three Meer Filters

The final MMAE Meer filter analysis indicates that a simple bank of 3 Meer filters performs well for a broad range of beam time constants. As the MMAE Meer performance in previous analyses seemed insensitive to variations in the beam time constant, the author chose filter 1, 2, and 3 beam time constants of 20, 200, and 2 seconds, respectively, for this test. This choice not only represents two orders of magnitude spread in the beam time constant, it stresses the sampling rate. The true beam time constant was set at a constant value of 70 seconds, as this is about halfway between 20 and 200 on a log scale. Figure 5.9 is the mean error history +/- 1 standard deviation for the adaptive state estimate. This figure is also representative of the performance of the Meer filters with beam time constants of 20 seconds and 200 seconds. Figure 5.10 is the performance

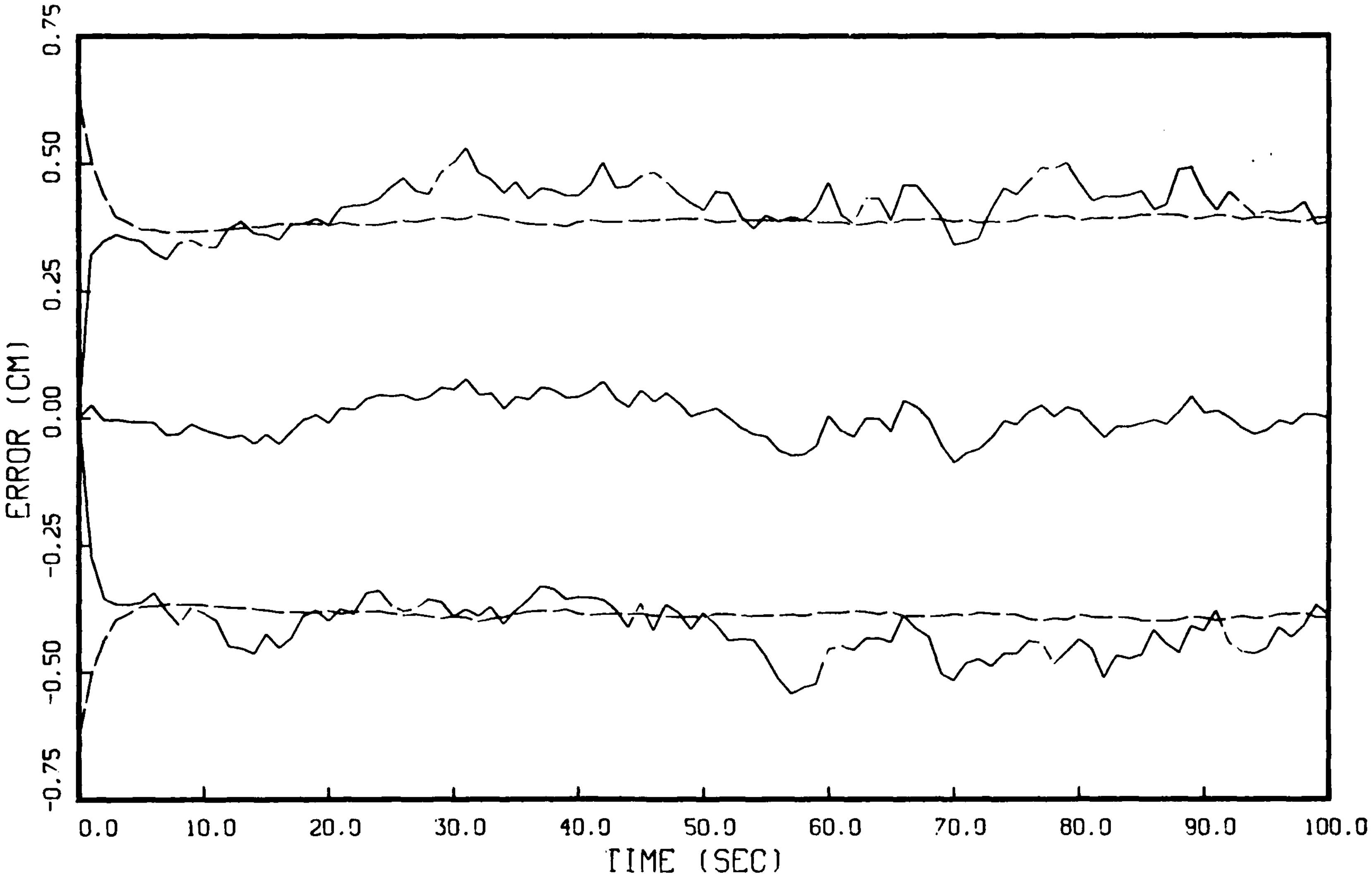


Figure 5.9 Adaptive State Estimation Error
True Tau = 70

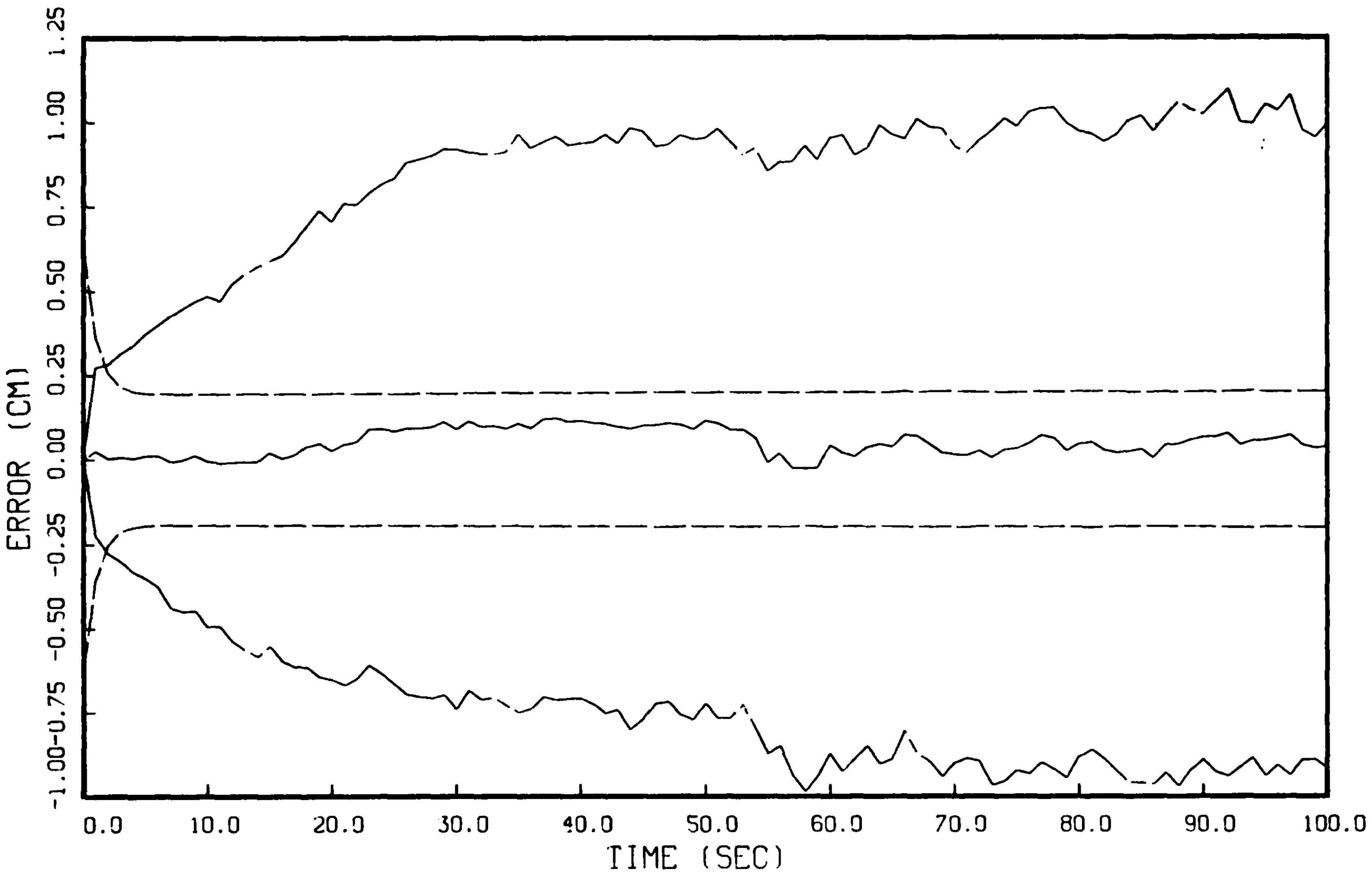


Figure 5.10 Meer Filter #3 State Estimation Error
Filter Tau = 2 True Tau = 70

of the Meer filter with beam time constant of 2 seconds. Of significance in Figure 5.9 is that the mean error is near zero and the filter-computed standard deviation is close to the actual standard deviation. In Figure 5.10 we see that the mean error does not remain near zero and the filter predicted standard deviation is much less than the actual standard deviation. This is an indication of degraded filter performance due to a poor assumed beam time constant. Note that in Table 5.7, the RMS error for the Meer filter with beam time constant of 200 is lower than the RMS error for the other filters. This is due to the gain placed on the measurements through Equation (2-22) that is part of the weighting applied to the residuals in Equation (2-46). The influence of the large beam time constant on Equation (2-22) is inferred through Equation (5-1): see the last column of Table 5.7. Also note that the RMS error is slightly lower than the filter predicted standard deviation. Thus, this

Table 5.7 Stressed Filter Performance

$R = .5\ \mathrm{cm}^2$ $g = .2\ \mathrm{cm/sec}^{1/2}$ Depth = 1
True beam time constant = 70 seconds

	beam time constant (sec)	RMS error (cm)	square root of filter variance (cm)	$\sqrt{P_{ss}}$
Adaptive		.444874	.429922	
Meer#1	20	.433784	.370901	.63
Meer#2	200	.406193	.406536	2.00
Meer#3	2	.928334	.201014	.20
	70			1.18

filter is performing well. In contrast, the filter predicted standard deviation of the Meer filter with beam time constant of 2 is much lower than the RMS error. This filter is operating improperly. For this filter, the variance in steady state is much smaller as inferred by Equation (5-1). Thus, measurements in a filter with a low beam time constant receive correspondingly lower weight through Equation (2-22) and the filter relies on its internal dynamics model. When the filter dynamics model is incorrect (as with this filter) the state estimation suffers. The degraded state estimate results in higher residuals for actual signal events. Higher residuals result in lower weighting on the measurements through Equations (2-7), (2-38), (2-39) and (2-46). Consequently, the filter that underestimates the beam time constant diverges. Note that for the divergent filter (Meer filter #3), the predicted value is approximately equal to the steady state variance defined in Equation (5-1) for a filter based on a beam time constant of 2 seconds. This indicates that the measurements are being discarded as noise. As the higher residuals of the divergent filter also result in a low probability weight on the state estimates from that filter, the MMAE Meer is able to reject the influence of a filter that underestimates the beam time constant. Although the filter computed variance of the Meer filter with beam time constant of 20 seconds is lower than the steady state variance indicated by Equation (5-1) for a beam time constant of 20 seconds, its square root is lower than the RMS error. Therefore, the filter is making use of the incoming measurements; however, the weighting is not sufficient to track the beam precisely. The RMS error of the adaptive state estimate also exceeds its filter-computed error standard deviation. This is not surprising as the adaptive state estimate for this run uses the nominal beam time constant of 20 seconds rather than the true beam time constant of 70 seconds to propagate to the measurement time. Also, this run was conducted prior to the setting of the Meer filter lower probability bound from .1 to 0.0005. Therefore, the divergent filter impacts the adaptive state estimate to a greater degree than would normally be expected.

The exceptional performance of the Meer filter with beam time constant of 200 becomes a problem for the adaptive estimation process. The adaptive parameter estimate is quickly pulled past the correct beam time constant of 70 seconds toward the filter with beam time constant of 200, as shown in Figure 5.11. The upper limit on the adaptive estimate due to the 0.1 lower probability bound is 162 seconds. However, the influence of the Meer filter with beam time constant of 20 seconds results in a mean adaptive beam time constant of 125 seconds (averaged over the final 50 seconds of the simulation). This is a poor adaptive estimate, but not a surprising result considering the slow trend toward the Meer filter with the highest time constant

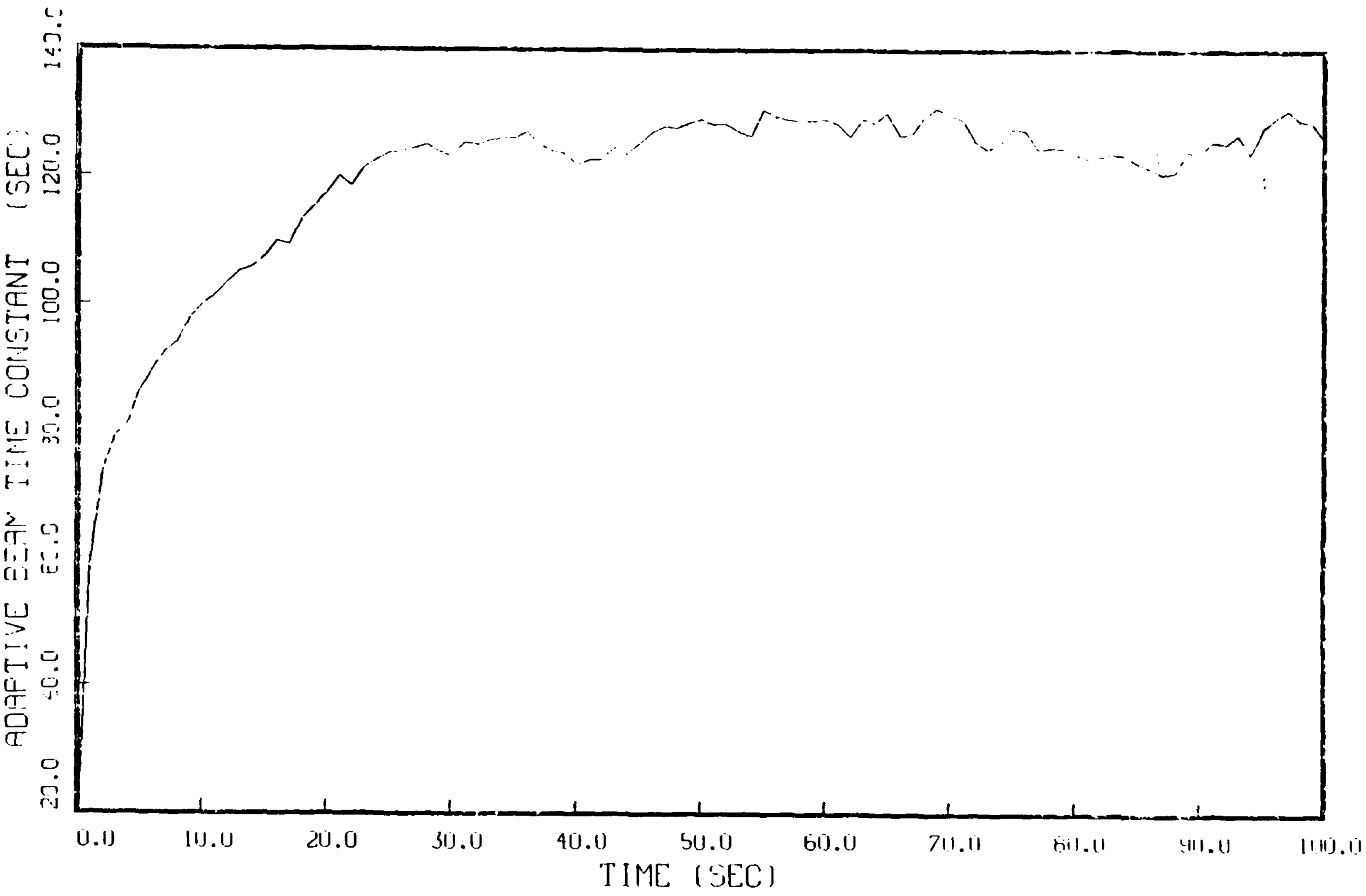


Figure 5.11 Mean Adaptive Beam Time Constant
True Tau = 70

in previous analyses. Thus, Meer filters with large beam time constants may dominate the adaptive beam time constant estimation process. It may be possible to reduce this influence by decreasing the dynamics driving noise term g as used in Equation (5-1). However, such detuning may have undesirable impacts on state estimation.

5.6 MMAE Meer Filter In Controller Loop

The results of the previous sections indicate that the Meer filter itself is rather robust with respect to the beam time constant. Then one asks why did Johnson [9] experience instability in the Meer filter when operated in the closed loop? This section explores that question and follows the attempts to stabilize the MMAE Meer filter in closed loop operation.

5.6.1 Beam Following Detector. The first step in exploring the closed loop operation was another review of the code to ensure there were no errors in the control loop. This review disclosed a significant error. In the implementation of the third order Gauss-Markov position process, no provision was made to ensure that the beam would remain over the detector. As the position of the target went to infinity, the controller forced the beam off the array to keep it on the target. Meer indicated [21:181], that the probability of a noise event outside the detector is zero. Therefore, the program set the probability of an assumed noise event to zero and the signal event probability became one, whenever an event occurred off the array. Meer did take into consideration the finite size of the detector on the generation of signal events [21:149]. However, in Meer's problem it was appropriate to set the array size much larger than the beamwidth and assume the probability of a signal off the array as negligible. For pointing of the beam at a highly dynamic target, such assumptions have to be reconsidered. As the initial target position and velocity were set to 0 cm and 5 cm/sec, the 10 cm detector used in previous studies was generally left behind in the first two seconds. This effectively reduced the Meer filter to a Snyder-Fishman filter with respect to signal events. The author believes this error cannot truly account for the instability of the estimator in the closed loop exhibited in Johnson's research [9], but it may have aggravated it. As a Snyder-Fishman filter, the Meer filter would weight each incoming signal event as an actual event. There would be no deweighting of the event due to a large residual. However, a noise event would still occur on the array so that the Meer filter would deweight the residual based on the size of the residual through Equations (2-7), (2-39) and (2-46). Therefore, noise events had neglible impact, while the filter was given full benefit of the information in signal events. Basically, the Meer filter was being told which events were noise induced and which events were signal induced. Therefore, this error cannot account for the instability problem. However, it probably did have impacts on the signal to noise (SNR) sensitivity and robustness studies done by Johnson [9] and his sensitivity studies involving variations in the number of expected signal events over the run.

The author considered three approaches to maintaining the beam on the array. The first approach was to increase the size of the detector. However, this would require a length of detector of several meters and therefore this approach was considered unrealistic. The next approach involved reestablishing the coordinate frame origin at each control update period to the estimated beam or estimated target location. This method was actually implemented, first based on estimated target position and then with estimated beam position. The results were virtually identical. Not only did this approach maintain the beam over the detector, it had a marked impact on the stability of the closed-loop system. Figure 5.12 depicts the mean error +/- 1 standard deviation envelope of the adaptive state estimate. However, it is also representative of the performance of the first and second Meer filters with beam time constants of 20 and 30 seconds respectively. Figure 5.13 depicts the performance of a Meer filter with a beam time constant of 10 seconds. The notable difference in this filter's performance is the steady state bias error. The wider range of filter beam time constants and a true beam time constant of 25 seconds was picked to emphasize the

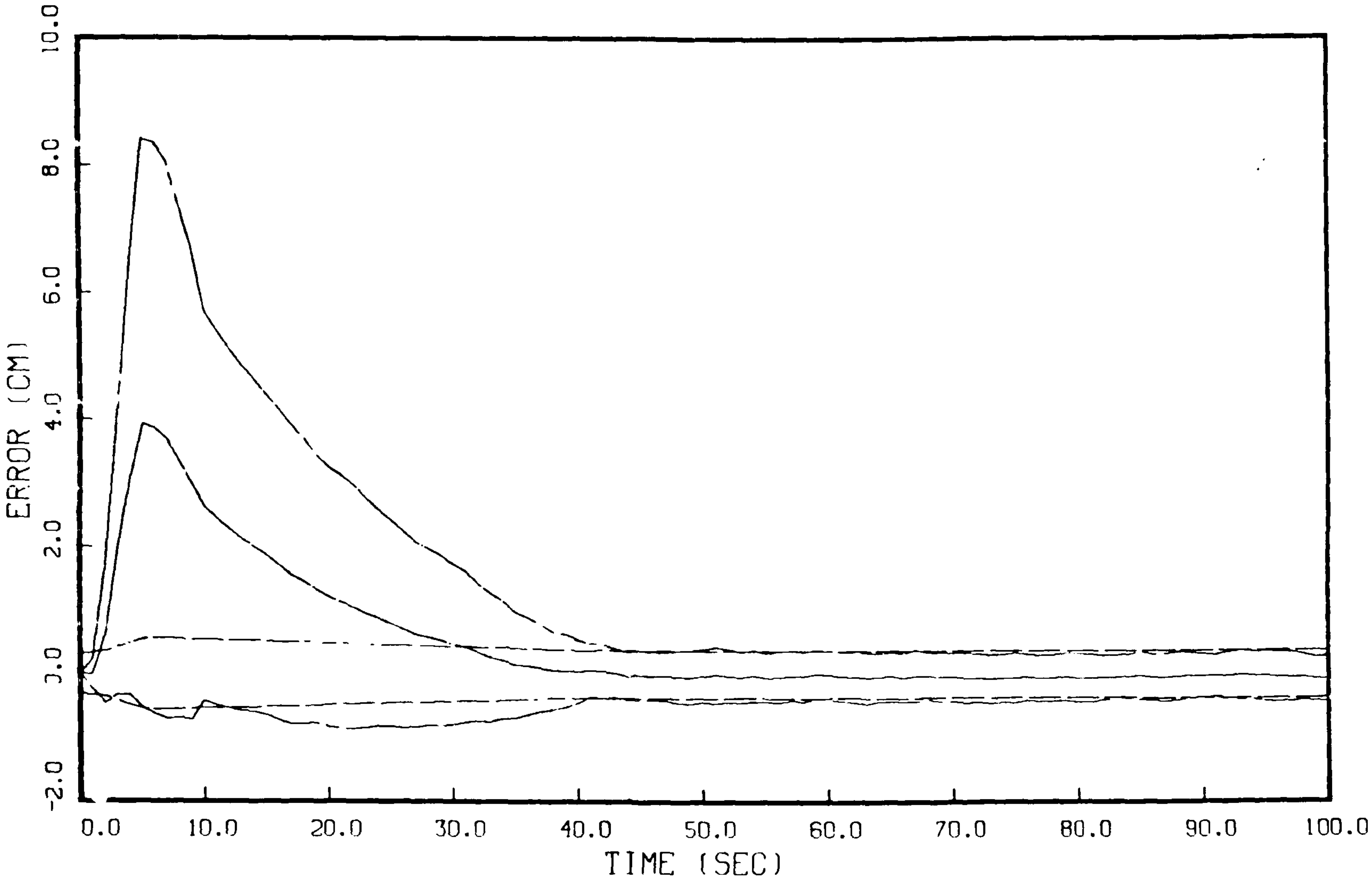


Figure 5.12 MMAE State Estimation Error; NPB Rotation
True Tau = 25

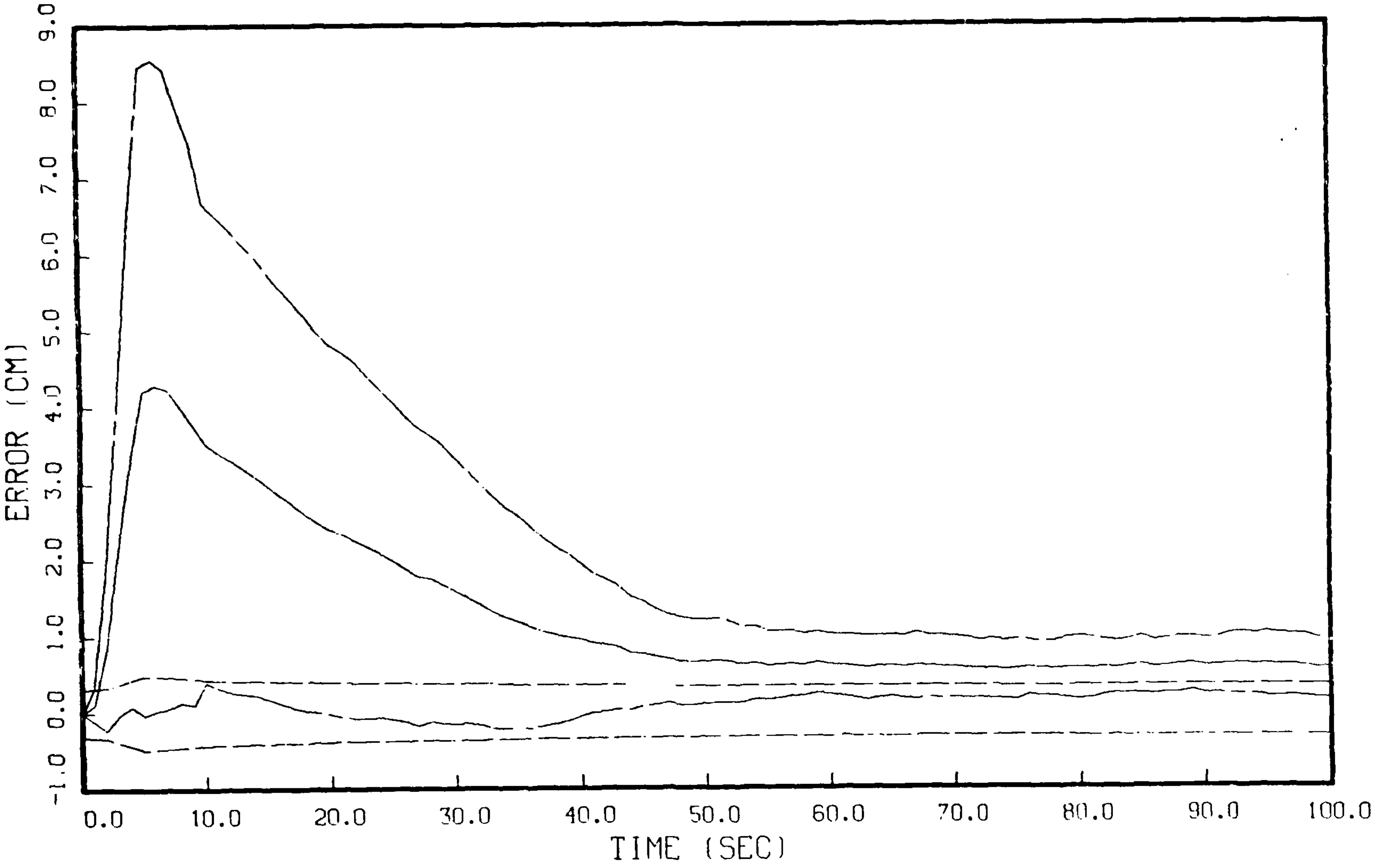


Figure 5.13 Meer Filter #3 State Estimation Error
True Tau = 25 Filter Tau = 10

robustness of this approach. In Johnson's research [9:119-123] a filter-assumed beam time constant of 20.0 seconds and a true beam time constant of 19.2 seconds or 21 seconds resulted in an unstable beam filter. What is important in Figure 5.12 is that the adaptive mean is near zero and the filter computed error standard deviation (as indicated by the two dashed lines that straddle zero) is close to the actual error standard deviation. Note that the MMAE filtered out the mean error of the third Meer filter (through its probability calculations). At the time of the actual runs for this investigation, the software had not been modified to output the individual probabilities nor the adaptive beam estimate. However, from the results of the previous sections, we may surmise that the MMAE estimate was dominated by the estimate of the Meer filter with the largest beam time constant. Important in Figure 5.13 is the slope of the mean error in steady state remains flat and the error standard deviation is well represented by the filter computed value, despite the presence of the steady state bias error. For none of these filters is there an indication of instability. Significant in these figures is the large initial error and the long transient before reaching steady state. For this run, as in all the runs, the initial target and beam positions were set to zero in both the truth and filter models. The initial target velocity state was 5 cm/sec and initial acceleration state was also zero. The initial error is due to the filter's lack of information during startup. When the initial sequence of measurements includes noise measurements, the initial estimate of the beam's position is very poor. Since all events now occur on the array, all measurements are deweighted through Equations (2-7), (2-39) and (2-46). Thus, it requires additional signal measurements for the filter to update toward the beam location. It should be noted that these results were obtained by supplying the MMAE Meer estimate to the MMAC developed by Johnson [9] with no retuning or adjustments. Figure 5.14 indicates how well this controller performed in tracking the beam onto the target. The upper line is the true RMS error in tracking error, whereas the lower line is the filter-computed RMS tracking error. Significant in this figure is that the true RMS tracking error is flat in steady state and the RMS error of the filter is close to this value. The large initial transient is due the initial transient in the beam filter estimate.

Although this approach appeared to return robustness to the closed-loop system, the author discontinued further development along this path. This was done because reinitializing the coordinate frame or the use of relative states versus absolute states in the real world represents rotating the entire NPB device toward the target. In a simulation this can be done easily, instantaneously and without error. In the real world this may be impractical for tracking a highly dynamic target due to the mass of the

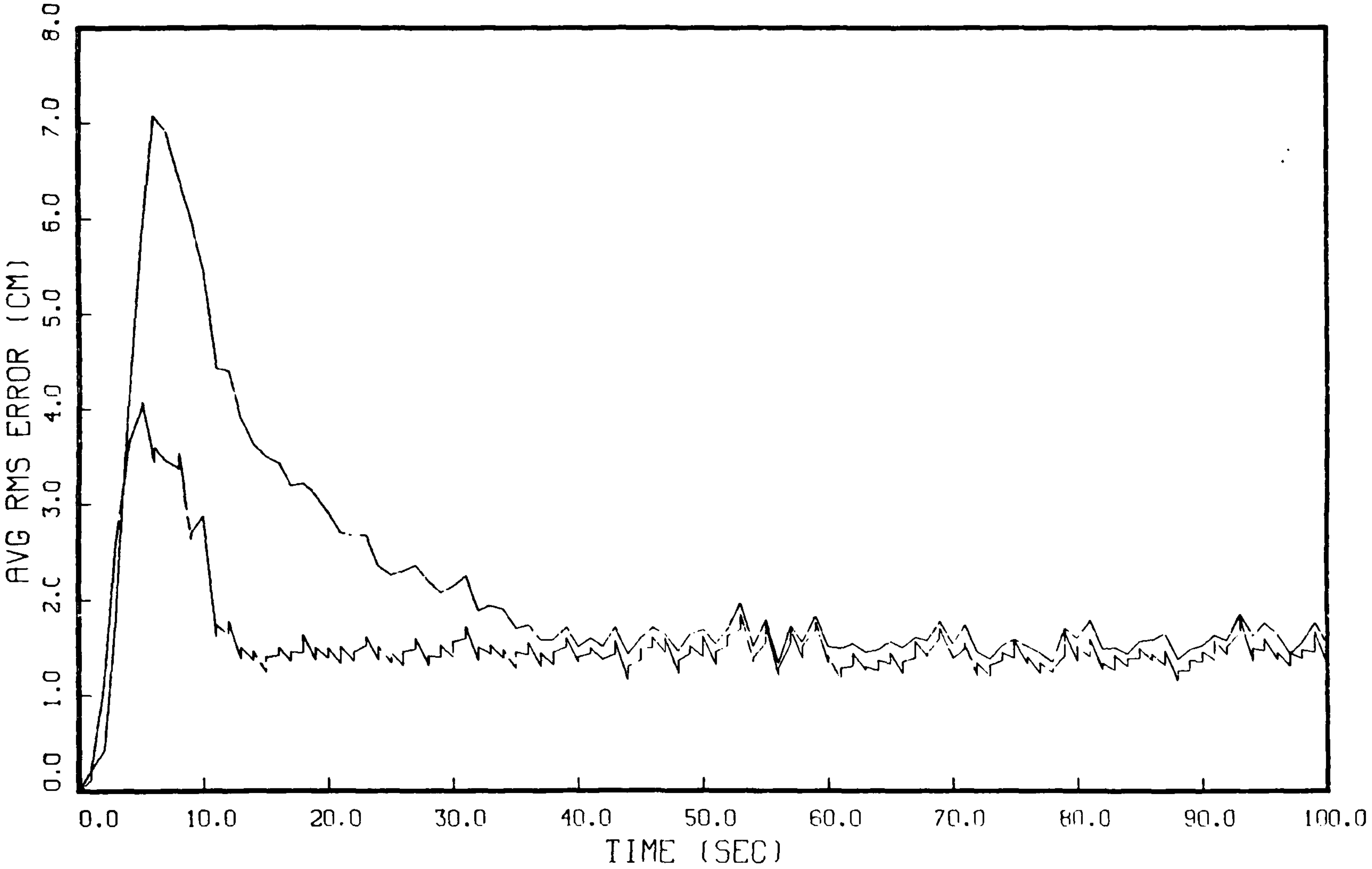


Figure 5.14 Tracking Error for NPB Rotation

NPB device. Thus, this approach may be more useful in providing slow coarse pointing adjustment. The third approach investigated a more practical method of tracking a highly dynamic target. Instead of moving the entire NPB device, the detector is moved. This would then provide vernier or fine tracking of the target. For simplicity (as this is a feasiblity study), the detector is repositioned instantaneously at the time of controller calculations. The new position of the detector is based on the estimated beam location halfway through the next controller update cycle. This was done in an effort to ensure that the beam would be over the detector so all the signal events would originate on the array. As discussed in Section 5.6.1 any signal events "occurring" off the array result in the filter being artifically informed that the event was a signal. In essence, we are no longer telling the filter which events are due to signal and which are due to noise.

This approach does not provide the same stability found with the first approach. Figures 5.15 and 5.16 indicate the performance of the MMAE Meer estimate and the best Meer filter performance which was provided by the Meer filter with the largest beam time constant. In this run, the range of beam time constants were set at 20, 20.4, and 19.6 seconds with the true beam time set to 20.2 seconds to minimize the growth of the actual error standard deviation. This was necessary so the entire error time history could be shown on one figure without deemphasizing fine detail, as

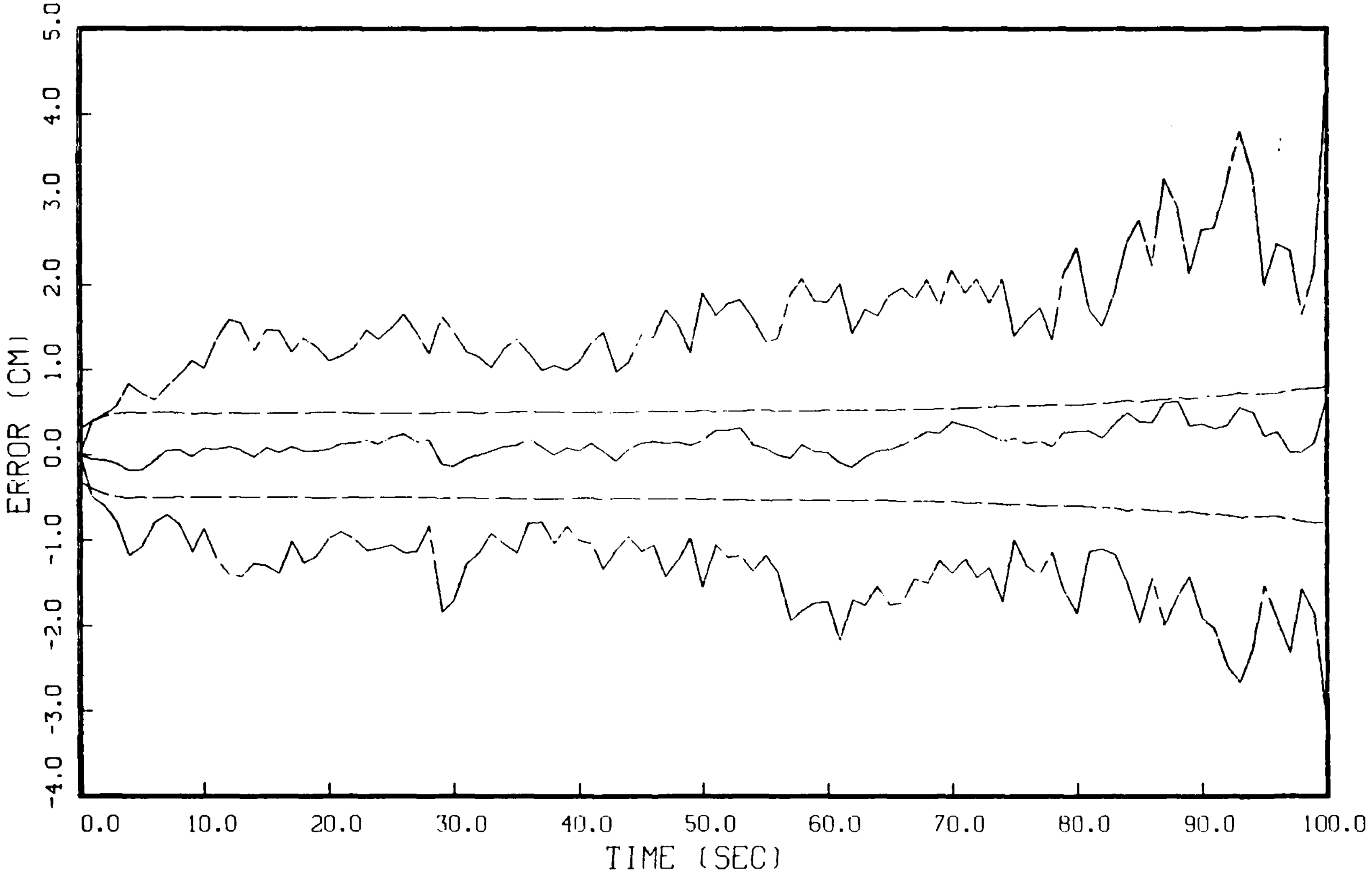


Figure 5.15 MMAE State Estimation Error
Detector Follows Beam

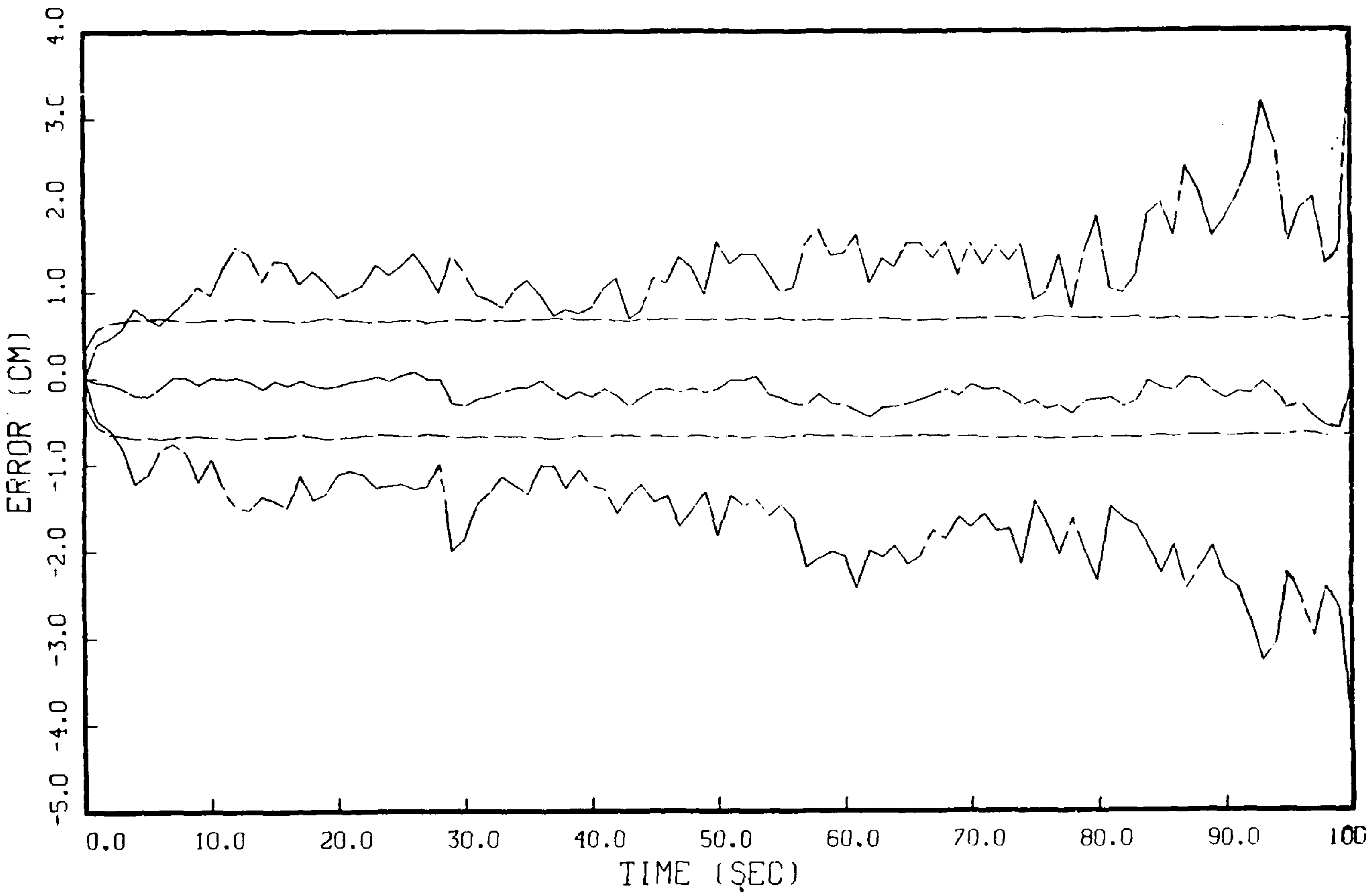


Figure 5.16 Meer Filter #2 State Estimation Error
Detector Follows Beam

would have occurred had the error standard deviation been large. Important in these figures is the poor filter computed error standard deviation, which is much less than the actual error standard deviation. In addition, the mean error does not remain near zero for the elemental Meer filter estimate, nor the adaptive estimate. Most important is that the actual error standard deviation is increasing with time. Both the drift of the mean error from zero and the growing error standard deviation indicate a divergent filter. The performance is also depicted in Table 5.8. Note that the square root of the filter predicted variance is again near the square root of the steady state variance of the beam calculated through Equation (5-1). Thus, the elemental Meer filters are not making effective use of the incoming measurements. Despite this, the MMAE Meer estimate was able to combine the poor elemental estimates into a marginally better estimate. As with the previous study,

Table 5.8 Filter Performance with Detector Following Beam

$R = .5\ \text{cm}^2 \quad g = .2\ \text{cm/sec}^{1/2} \quad \text{Depth} = 1$

True beam time constant = 20.2 seconds

	beam time constant (sec)	avg mean error (cm)	square root of filter variance (cm)	$\sqrt{P_{ss}}$
Adaptive		.148876	.555665	
Meer#1	20.0	.277048	.665026	.632
Meer#2	20.4	-.170792	.673304	.639
Meer#3	19.6	.752915	.656665	.626
	20.2			.636

this run was actually conducted prior to the modifications to the code necessary to allow the elemental probability histories to be sampled and output so their ensemble averages could be calculated. We can see from Figure 5.15 that the MMAE estimate is diverging, just as the elemental filters are diverging. There are two reasons for the elemental filter divergence. The first reason is the deweighting of the measurements by the Meer filter as has already be discussed. However, as we previously noted, Johnson [9] experienced divergence even when the Meer filter was fully weighting signal measurements and providing neglible weight to noise measurements. The second reason is due to the physical nature of the problem of tracking an object that can move to infinity. Recall from Section 3.1.1 that our target model involves a double integration process. To track this target, the controller must force the beam state, as modelled in this simulation, to infinity. As the beam state grows large, the propagation error to the next beam state due to a mismodelling of the beam time constant also grows large as depicted in Figure 5.17. In this figure the upward ramping line is the propagation error committed by a filter with a time constant of 20.8 seconds. The downward ramping line is the propagation error committed by a filter with a time constant of 19.2 seconds. The propagation equation used is Equation (2-17) with a propagation time interval of 1 second and a control input of 10. The control input chosen represents a nominal value

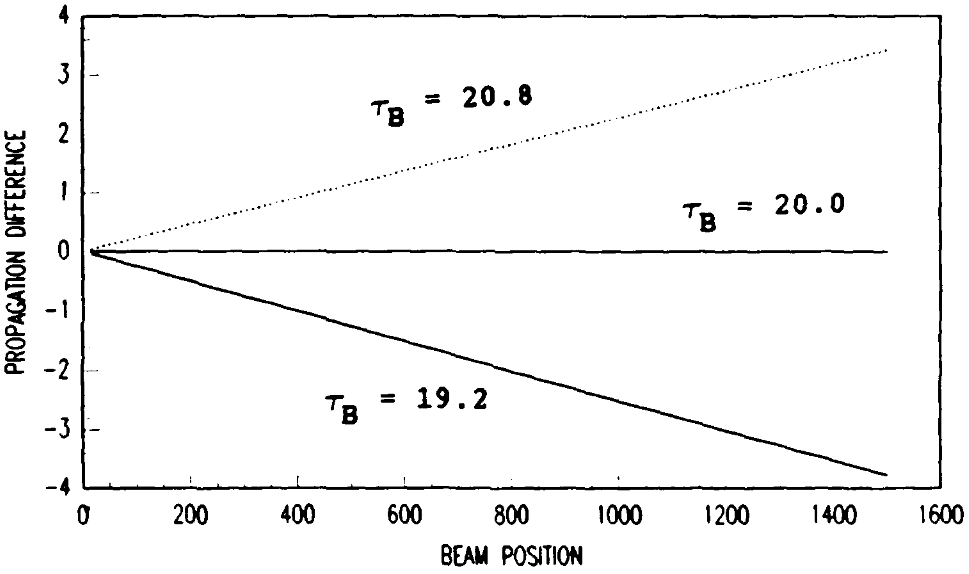


Figure 5.17 Propagation Error versus Beam Position

necessary to track the target. Although a control input can be applied to eliminate the error (a control input equal to the beam state) this would not in general track the target (unless the target position is reset to zero as in the first approach). As seen in Figure 5.17, the propagation error eventually exceeds the beamwidth and so signal measurements become deweighted just as noise. Therefore, it would seem that the ultimate solution to the stability problem involves rotating the NPB device or using relative states versus absolute states. However, it may still be possible to extend operations beyond propagation errors that exceed the beamwidth using the MMAE techniques. However, the number of elemental filters would be dependent on the beam position. The concept is to have sufficient elemental filters such that the propagation error between adjacent filters (in parameter space) does not exceed the beamwidth. The elemental filters near the true beam time constant would make effective use of the incoming measurements and remain stable. The other filters would eventually become divergent. However, the divergent filters could be reset based on the adaptive estimate of the nondivergent filters. To do this, we would need to detect when an elemental is divergent. Figure 5.18 depicts the mean residual performance of the Meer filter with a beam time constant of 19.6 seconds. Significant here is positive growth in the residuals, indicating the negative propagation error in this filter's estimate of the true beam position, as indicated by

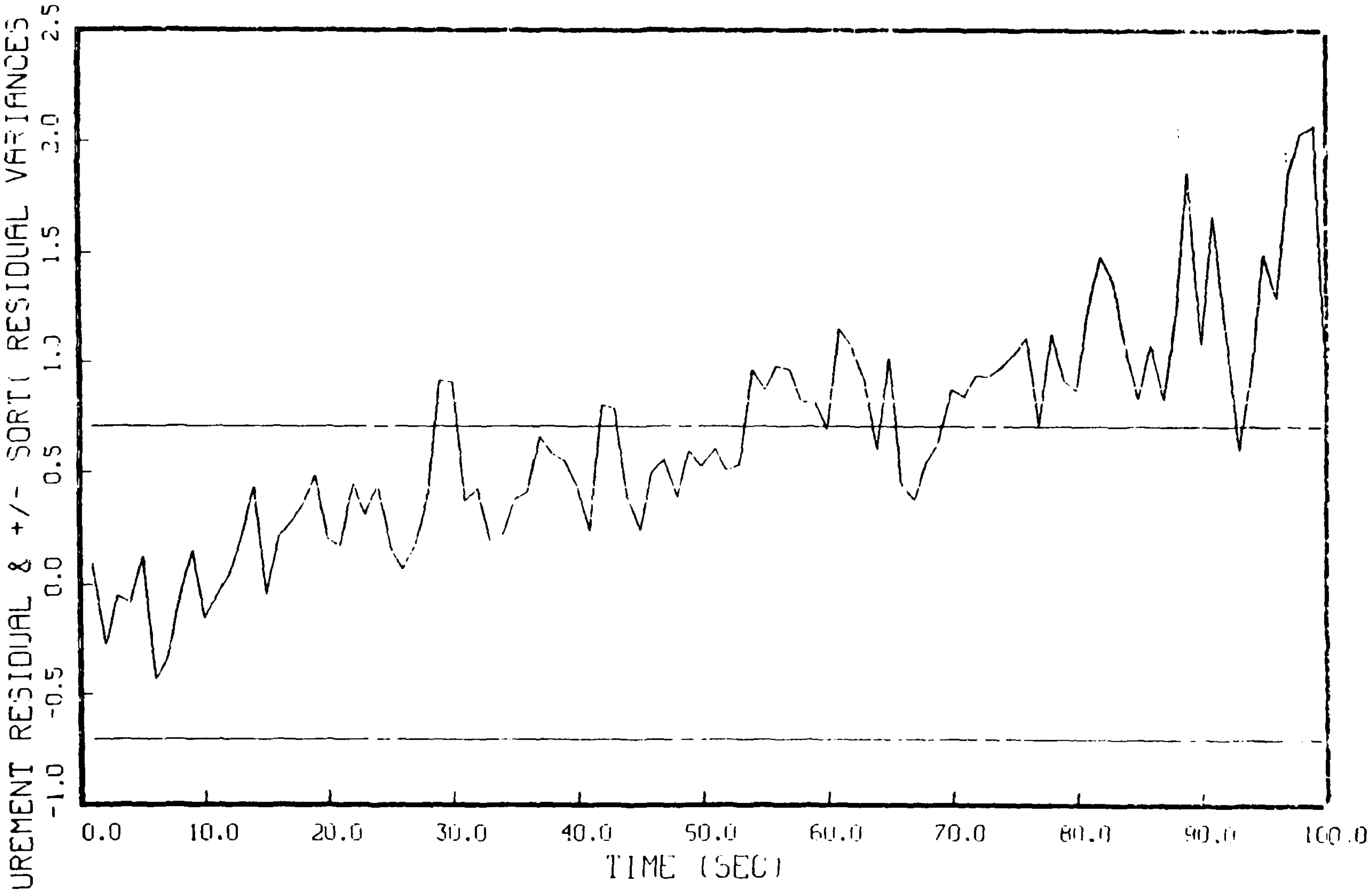


Figure 5.18 Mean Residual Meer Filter #3
Filter Tau = 19.6 True Tau = 20.2

Figure 5.17. Thus, through residual monitoring we may be able to detect the divergence of the filter.

5.6.2 Divergence Detection and Filter Reset. As the elemental Meer filters would surely be diverging, it was necessary to design a method of determining divergence and then to reset the filters. The author chose to detect divergence through residual monitoring. As a filter diverges, the mean of the residuals will increase, as seen in Figure 5.18, or their variances will increase, whereas the filter expectation of the residual squared values given by (scalar measurement) [16:229],

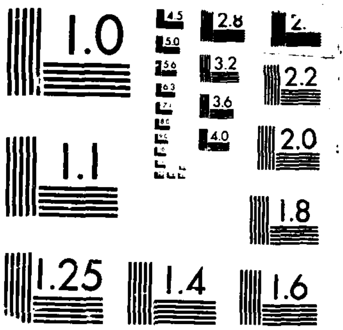
$$E\left\{r^2(t_i)\right\} = H(t_i)P(t_i^-)H(t_i) + R(t_i) \qquad (5\text{-}2)$$

would remain at a steady state value corresponding to the steady state value of the variance $P(t_i^-)$. What is needed is a method to determine when a filter has diverged, so that it can be reset. Since we have assumed the residual sequence is a zero-mean Gaussian stochastic process, we will detect divergence with the likelihood function [16:230],

$$L_N(t_i) = c(t_i) - 1/2 \sum_{j=i-N+1}^{i} r^2(t_j)/\,\sigma^2(t_j) \qquad (5\text{-}3)$$

where N is the number of residuals being processed, c is an

AD-A194 622 TRACKING AND CONTROL OF A NEUTRAL PARTICLE BEAM USING 3/3
MULTIPLE MODEL ADAP..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI..
UNCLASSIFIED L J HARAMBASIC DEC 87 AFIT/GE/ENG/87D-23 F/G 20/7 NL



MICROCOPY RESOLUTION TEST CHART

UREAU OF STANDARDS-1963-A

arbitrary, slowly varying value and

$$1/\sigma^2(t_j) = [H(t_j)P(t_j^-)H(t_j) + R(t_j)]^{-1} \qquad (5\text{-}4)$$

For this design c is set to zero. The output of the likelihood function is checked against a user selected threshold that indicates divergence. For this study the design consisted of individual divergence detectors (likelihood functions) for each filter and reset a divergent filter individually when a threshold of 9 (three standard deviations squared) was exceeded. As a filter within the bank diverges, its probability should drop relative to the nondivergent filters. The impact of this filter on the adaptive state estimate then becomes small. Thus, for this study, the adaptive state estimate is used to reset a divergent filter. Properly, the divergence check should occur prior to the calculation of the adaptive state estimate so the divergent filters can be removed from the estimation process. However, this correction was not incorporated into this study. This design used a value of N = 5 in Equation (5-3). This value was chosen by determining the approximate time that the state estimate of an elemental filter with beam time constant of 19.2 seconds differed from the true state by three beam halfwidths when the true beam time constant is 20.4 seconds. The value of N was then set so resetting of the filter occurred shortly after that time. Although crude, the author deemed this adequate for a signal to noise ratio of ten. For lower signal to noise ratios, a larger value of N would be more appropriate. A larger value of N would reduce the chance of improperly resetting the filters based on consecutive noise events. Recall that noise events are equally likely over the entire array rather than clustered near a mean regardless of the performance of the filter. Recall also that signal events can also have large residuals for a nondivergent filter, although the arrival rate is low for such events. Unfortunately, a larger value for N would delay the detection and reset of a divergent filter. Thus, the adaptive estimate may suffer. Care must also be taken in selecting a threshold for declaring divergence in a noisy environment. Since the noise rate is constant over the array, noise events at the array edges have the same probability of occurring as noise events near the beam. Also, we have defined the minimum appropriate detector length in Section 2.2.1 as 12 beamwidths. Thus, most of the noise events are considered to have large residuals even for a nondivergent filter. Therefore, a low threshold can be exceed with a few noise events despite a large value of N. This drives a desire to minimize the array length to limit the size of residuals from noise events, as well as, the rate of noise events.

5.6.3 MMAE Meer Baseline. With the Meer probability bound set to .0005 as mentioned in Section 5.2 and the filter divergence detection and reset processes in place, three runs were made to establish a baseline of Meer filter state estimation and parameter estimation performance in a closed loop. The controller for this closed loop baseline was designed with the weighting matrices X_{11} = 100 (target tracking error weight), X_{55} = 10 (pseudointegral state weighting), and U = 1 (control weighting) used by Johnson [9]. Since the desire of this section is to investigate the MMAE Meer operation and not the controller operation, the controller was provided the true beam and target states for these runs. The Meer filters 1, 2, and 3 beam time constants were 20, 20.8, and 19.2. The first run had a sinusoidal beam time constant with an initial value of 20 seconds and completed one cycle in 100 seconds as in Section 5.4. Since preliminary runs indicated that the adaptive parameter estimation process was functioning properly, the propagation of the adaptive state estimate to measurement times used the adaptive parameter estimate provided by the filter. Figure 5.19 depicts the mean error of the adaptive estimate. Figure 5.20 depicts the response of the Meer filter based on a beam time constant of 20 seconds. It is representive of the performance of the elemental filters. Both the MMAE Meer filter and the elemental Meer filters appear to be underestimating their own errors. Also the actual error standard deviations are increasing. This indicates the filters are continuing to diverge. As we have provided the true states to the controller, we have isolated

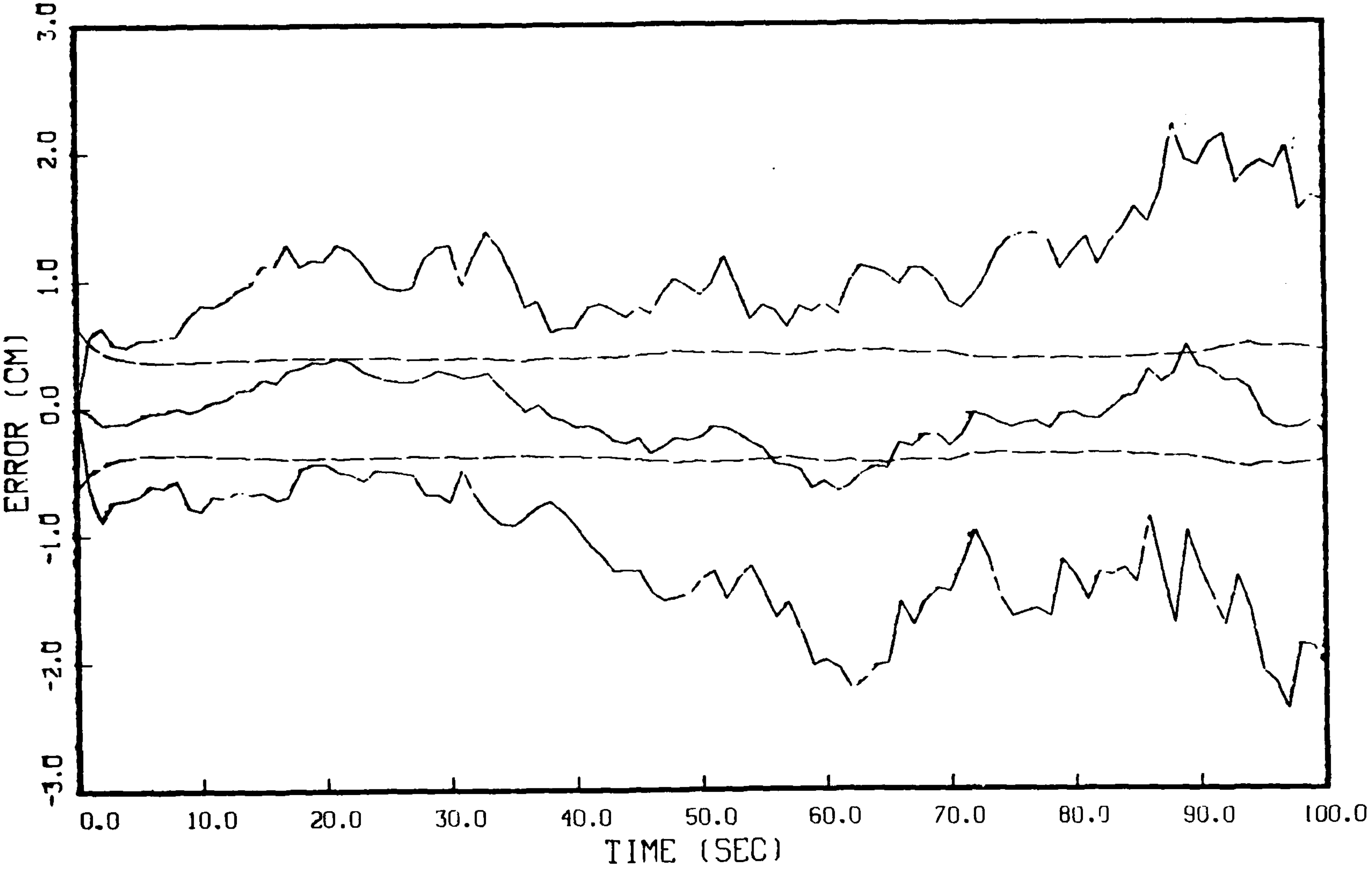


Figure 5.19 MMAE State Estimation Error; Oscillating True Tau = 20 + .8 sin(2πt/100)

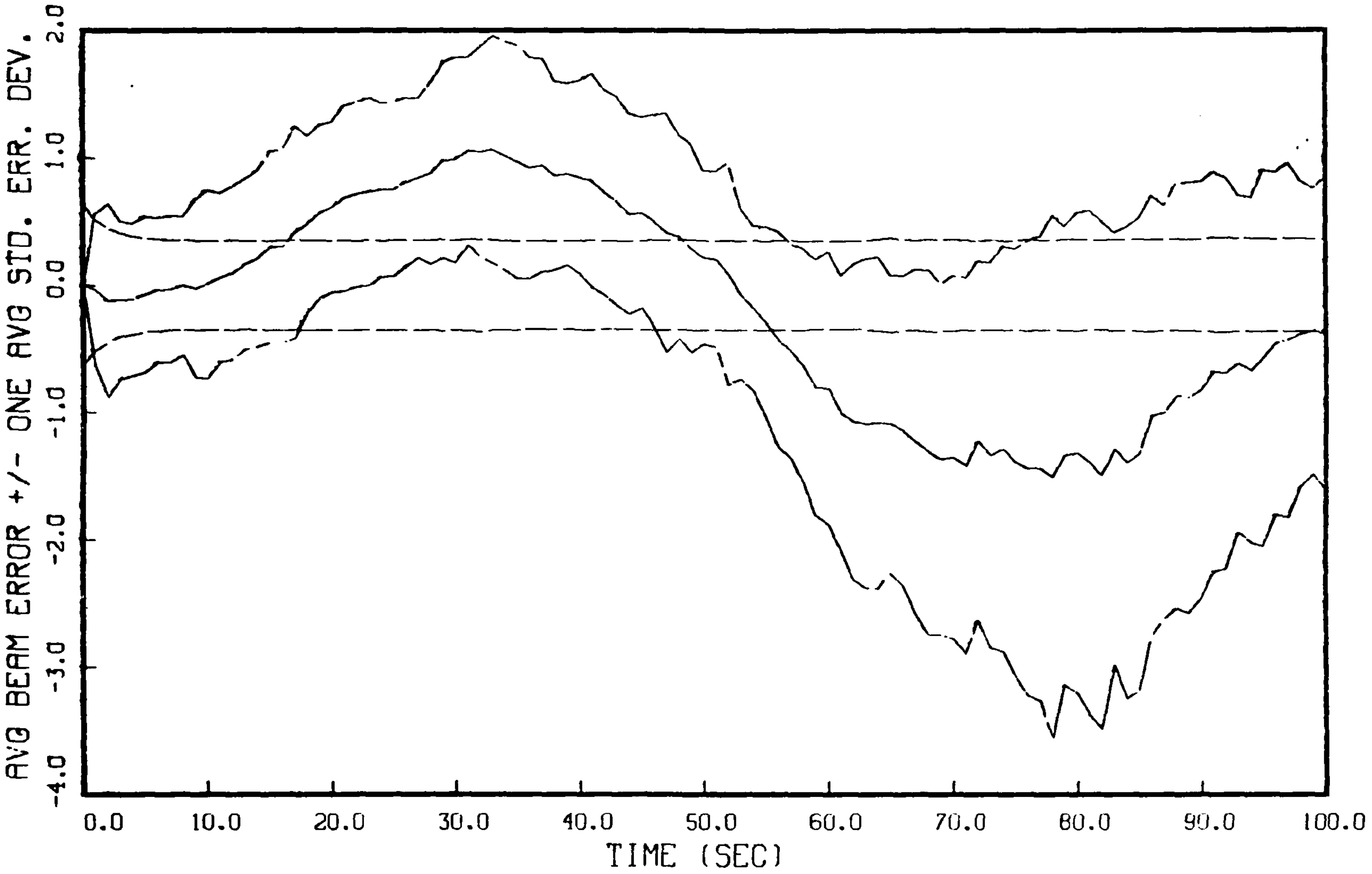


Figure 5.20 Meer Filter #1 State Estimation Error
Filter Tau = 20; Oscillating True Tau

closed loop beam errors to the beam filter. Note that the parameter estimation performance seen in Figure 5.21 has improved from the open loop case as indicated by the general result of Figure 5.3. The dots on Figure 5.21 serve as reference points of the correct beam time constant. As can be seen, the adaptive beam time constant is a reasonably good estimate of the true beam time constant, with some lag. The adaptive state estimate, in this case, is far better than any of the elementals as one might expect.

The second baseline run used a steady true beam time constant of 20.4 seconds. Figures 5.22 and 5.23 depict the MMAE performance and a representative performance of the elemental estimators. Again divergence in both figures is noted. Figure 5.24 shows the adaptive parameter estimate. Although the mean adaptive estimate is close to the correct value of 20.4 seconds, it is subject to variation. Again, this is reasonable parameter estimation performance. The mean probability time histories are shown in Figures 5.25, 5.26, and 5.27. Significant in these figures is that the mean probability is lowest for the Meer filter with beam time constant that is the farthest from the true beam time constant. Overall the parameter estimation performance of the MMAE Meer filter for a controlled beam is good.

The third run used the nominal beam time constant of 20 seconds. This value was chosen to examine the effect of incorporating the MMAE Meer filter structure with divergence detection and filter reset processes when the true beam time

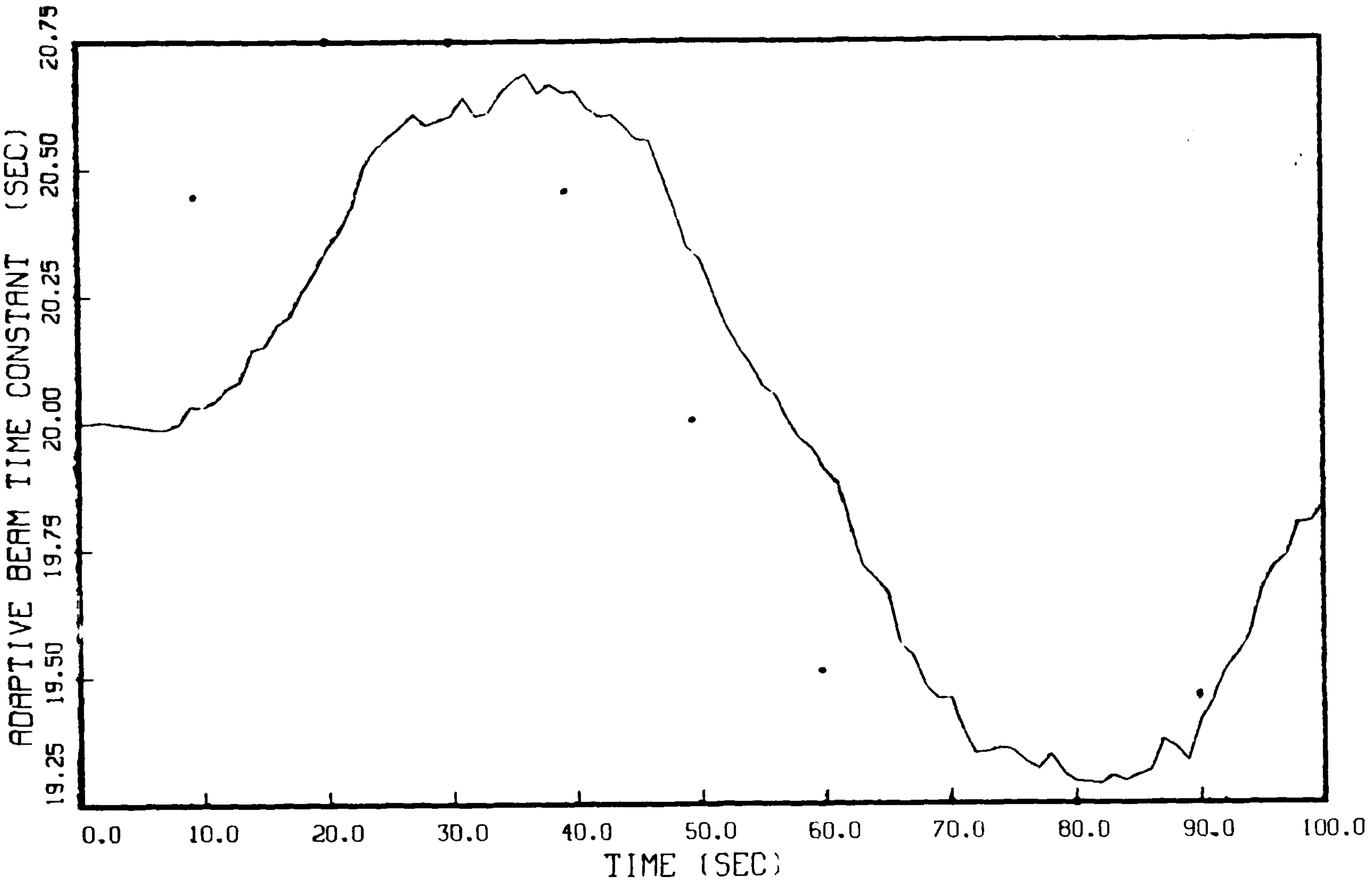


Figure 5.21 Adaptive Beam Time Constant; Oscillating True Tau = 20 + .8 sin(2πt/100)

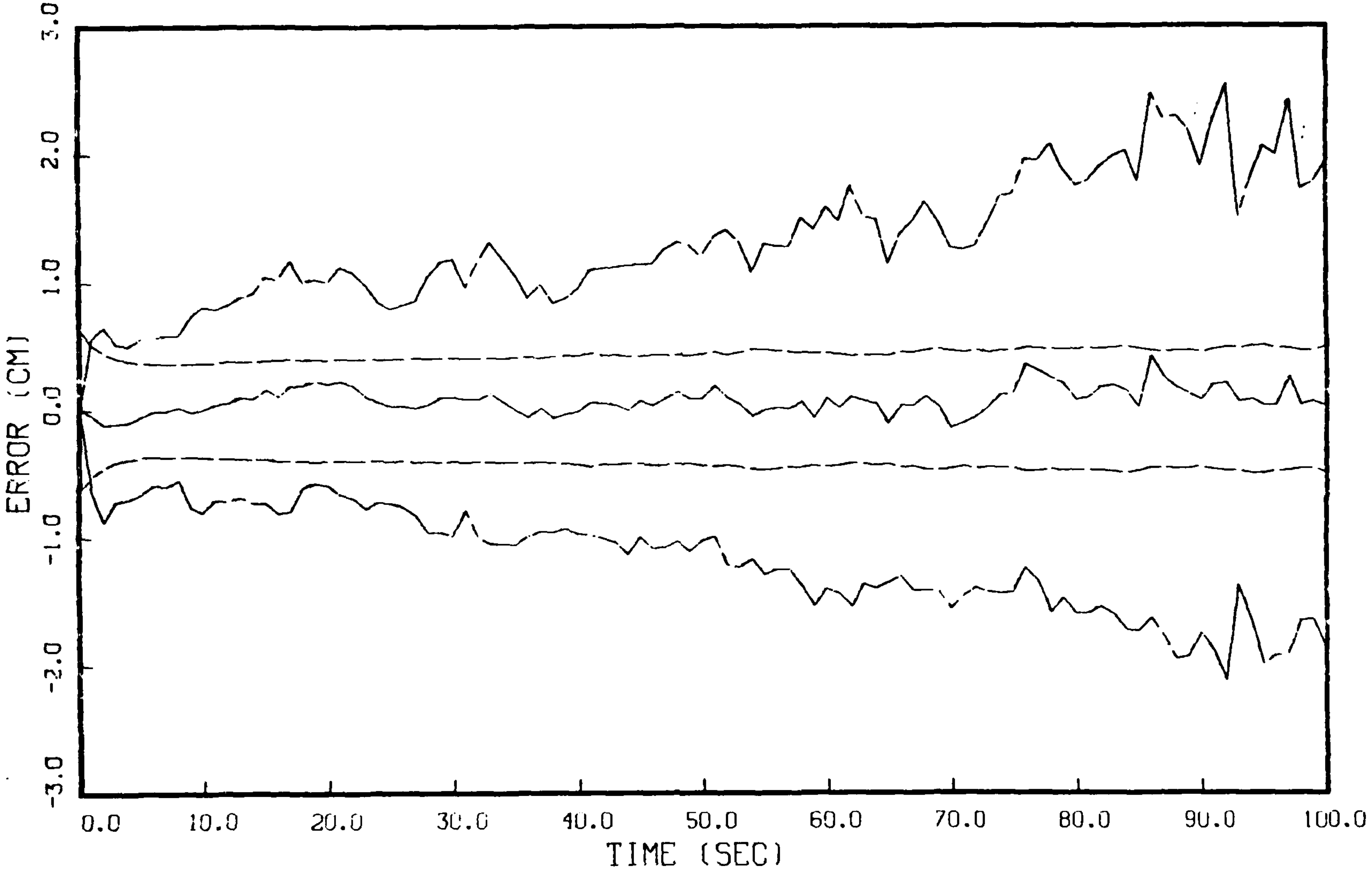


Figure 5.22 MMAE State Estimation Error
True Tau = 20.4

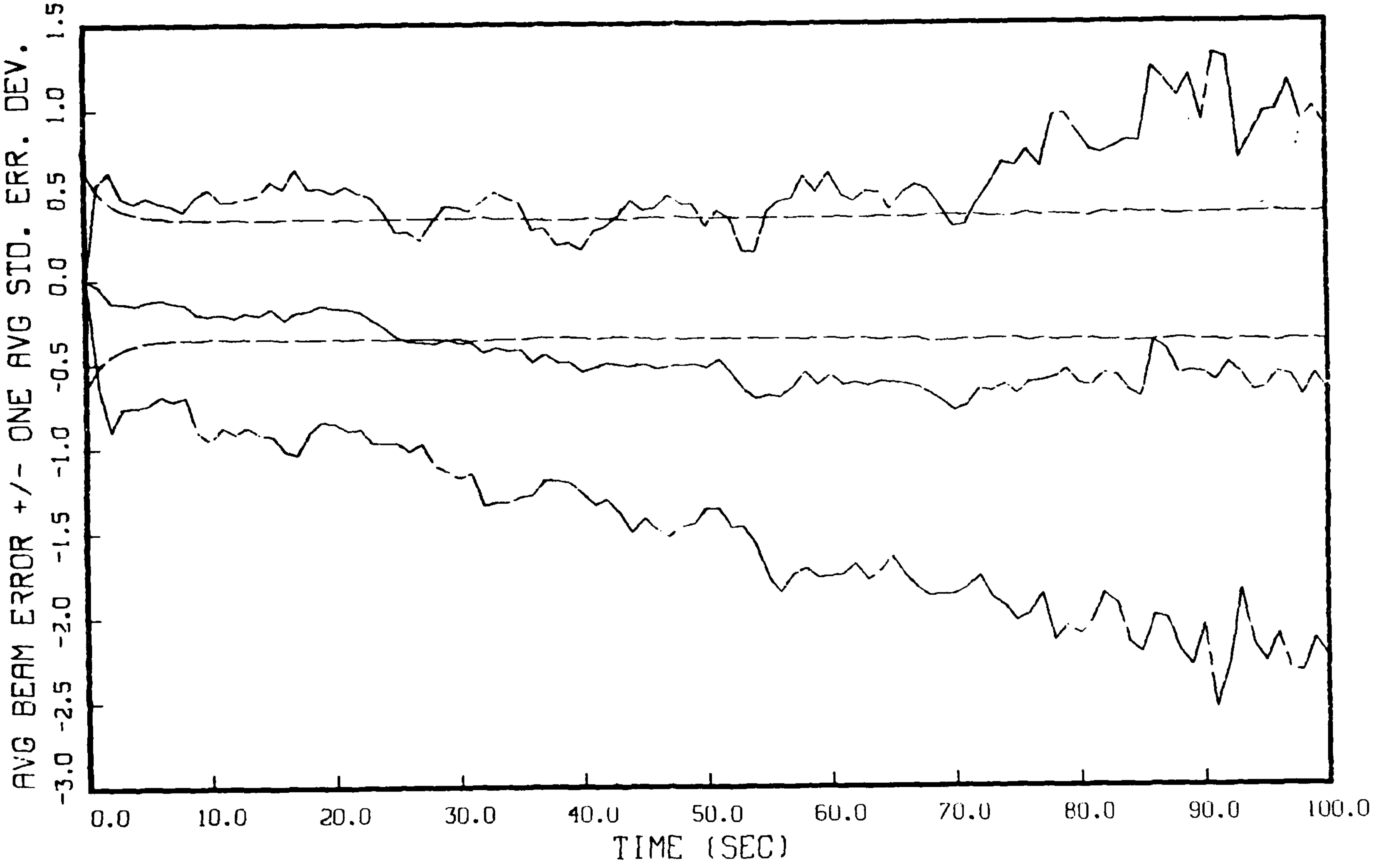


Figure 5.23 Meer Filter #2 State Estimation Error
Filter Tau = 20.8 True Tau = 20.4

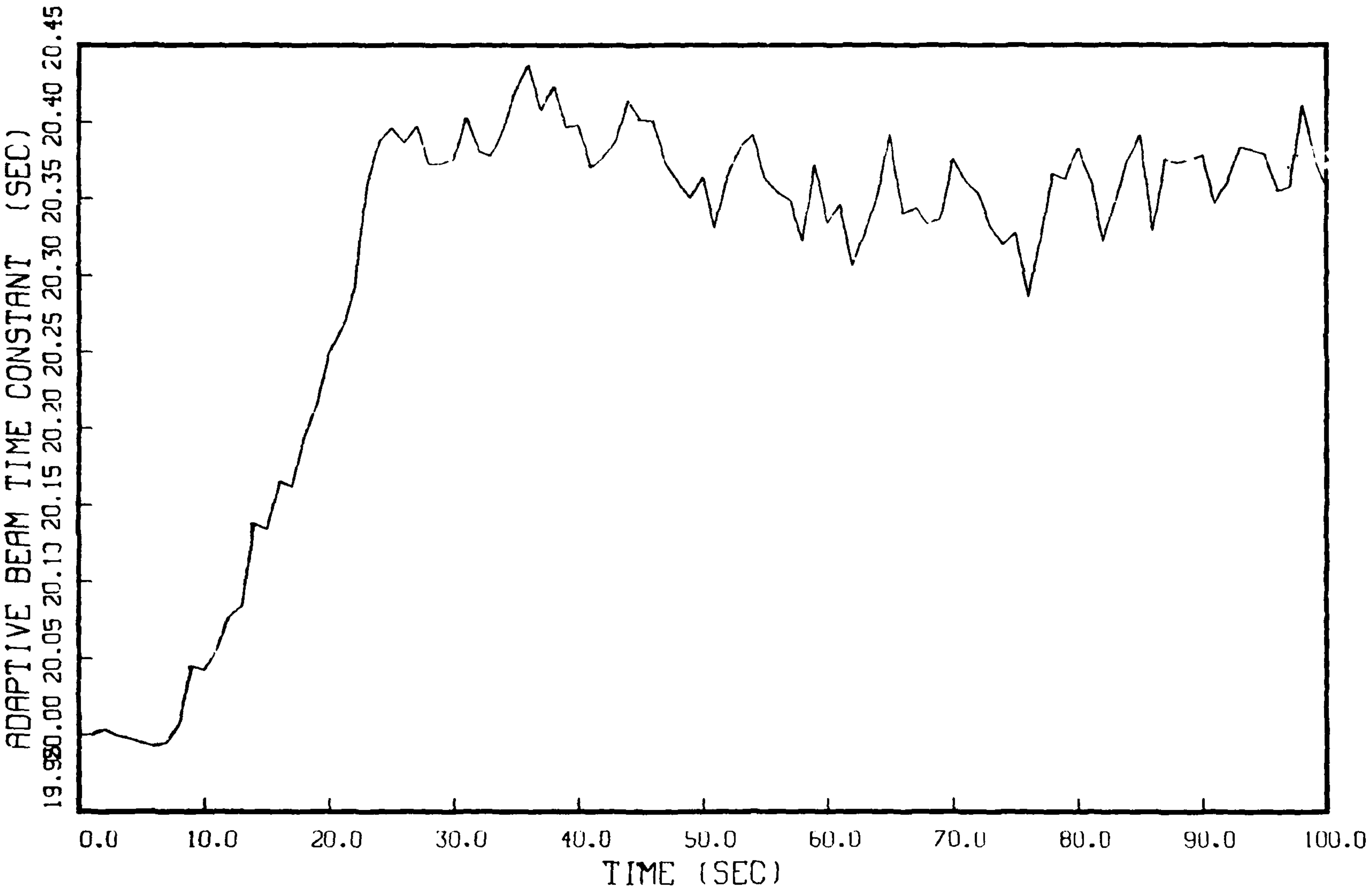


Figure 5.24 Closed Loop Mean Adaptive Beam Time Constant
True Tau = 20.4

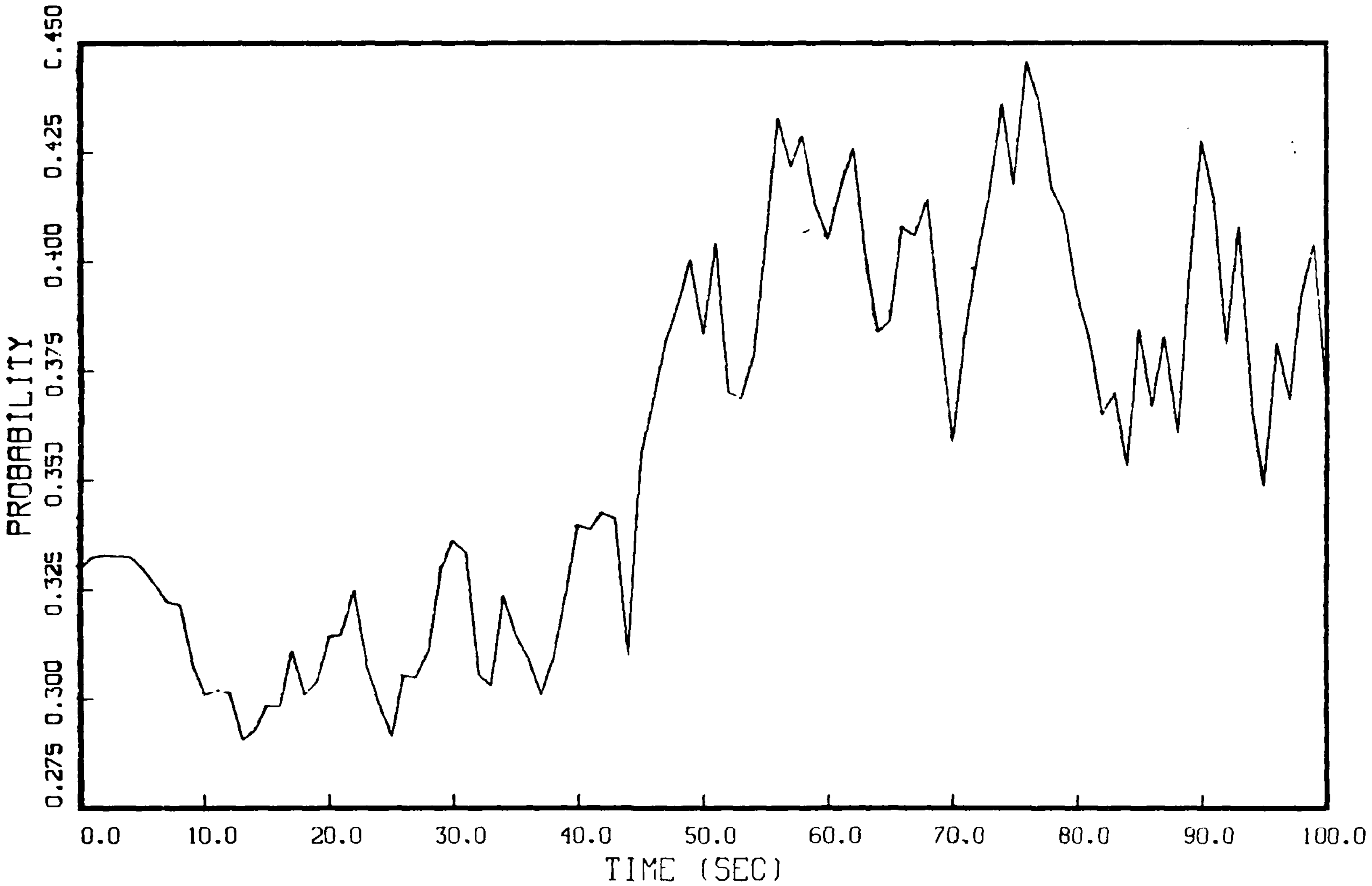


Figure 5.25 Closed Loop Probability History Meer Filter #1
Filter Tau = 20.0 True Tau = 20.4

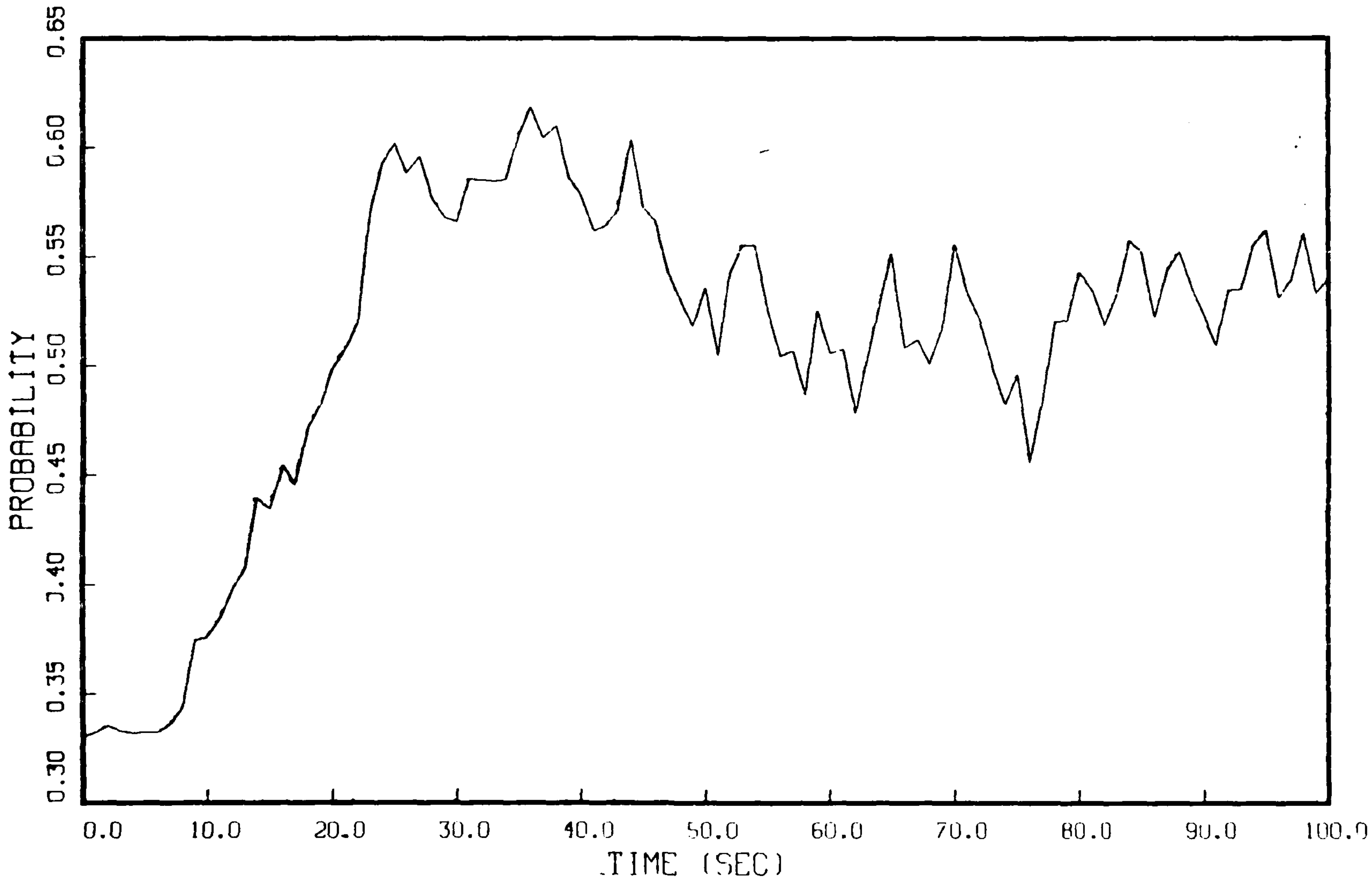


Figure 5.26 Closed Loop Probability History Meer Filter #2
Filter Tau = 20.8 True Tau = 20.4

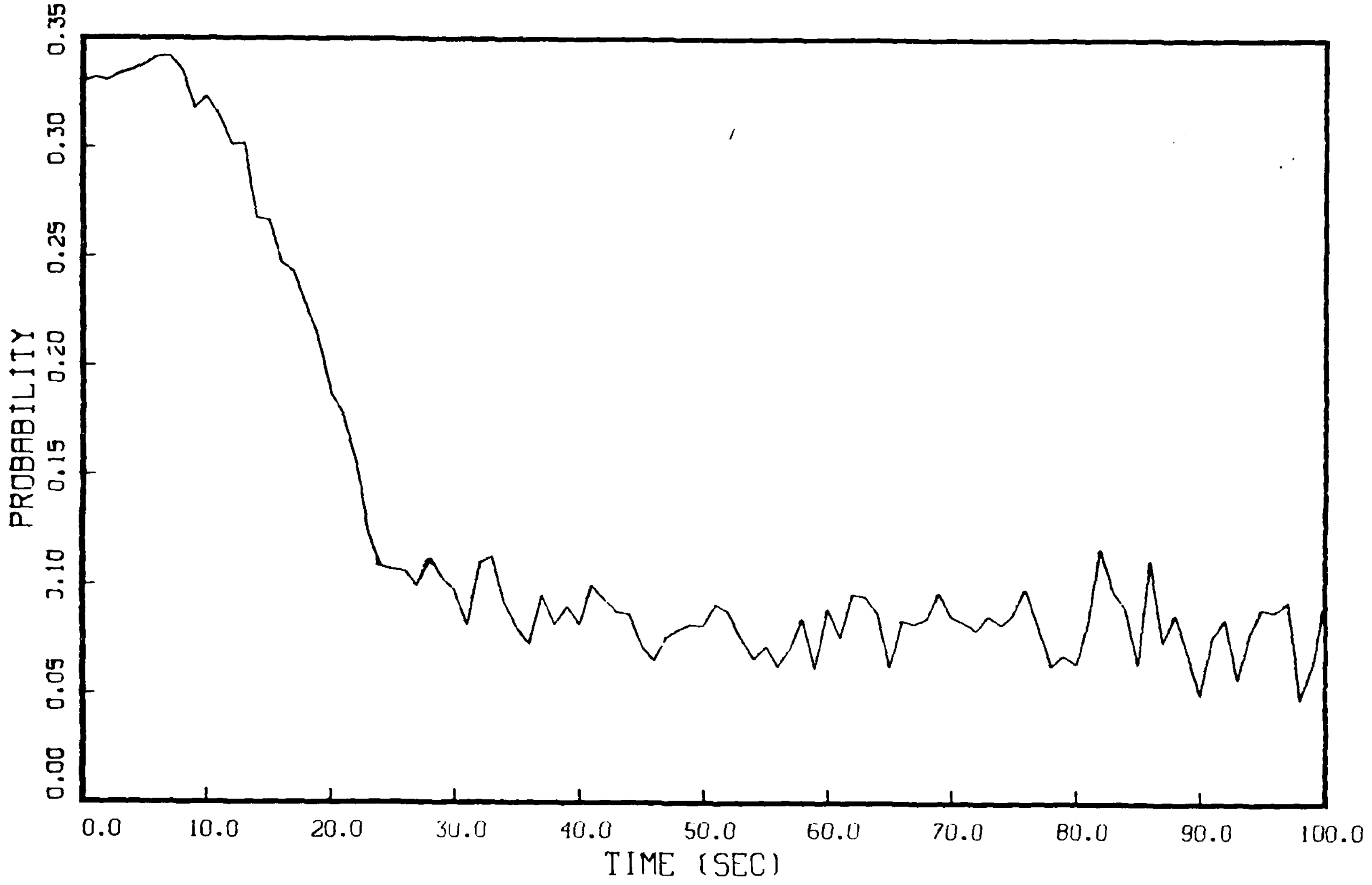


Figure 5.27 Closed Loop Probability History Meer Filter #3
Filter Tau = 19.6 True Tau = 20.4

constant matches one of the filter-assumed beam time constants. Figure 5.28 shows the performance of the MMAE. As before, the actual standard deviation exceeds the filter computed standard deviation. In addition, the error standard deviation of the adaptive estimate is growing. This is rather surprising given that one of the Meer filter assumed beam time constant matches the true value. However, Figure 5.29 indicates that even the filter that matches the beam time constant is underestimating its error. It might appear that the filter requires some tuning. However, it should be noted that the filter dynamics model matches the true dynamics model and the filter parameter values R and g match those of the truth model. Figure 5.30 is the RMS beam error time histories (true and filter-computed) of the Meer filter with the correct beam time constant. The upper solid line is the true beam error committed by this filter. The lower broken line is filter-computed one standard deviation. This figure clearly indicates that the filter is under-estimating its error. Since the actual error of this filter in growing with time the filter is divergent. As we are supplying the controller with the true states, we have again isolated the error to the beam filter. As discussed previously, the filter once again appears to be under-weighting its measurements, resulting in divergent behavior in state estimation. Examining the mean adaptive beam time constant in Figure 5.31, we note that the adaptive estimate does remain near the true value of 20. Also, in Figure 5.32

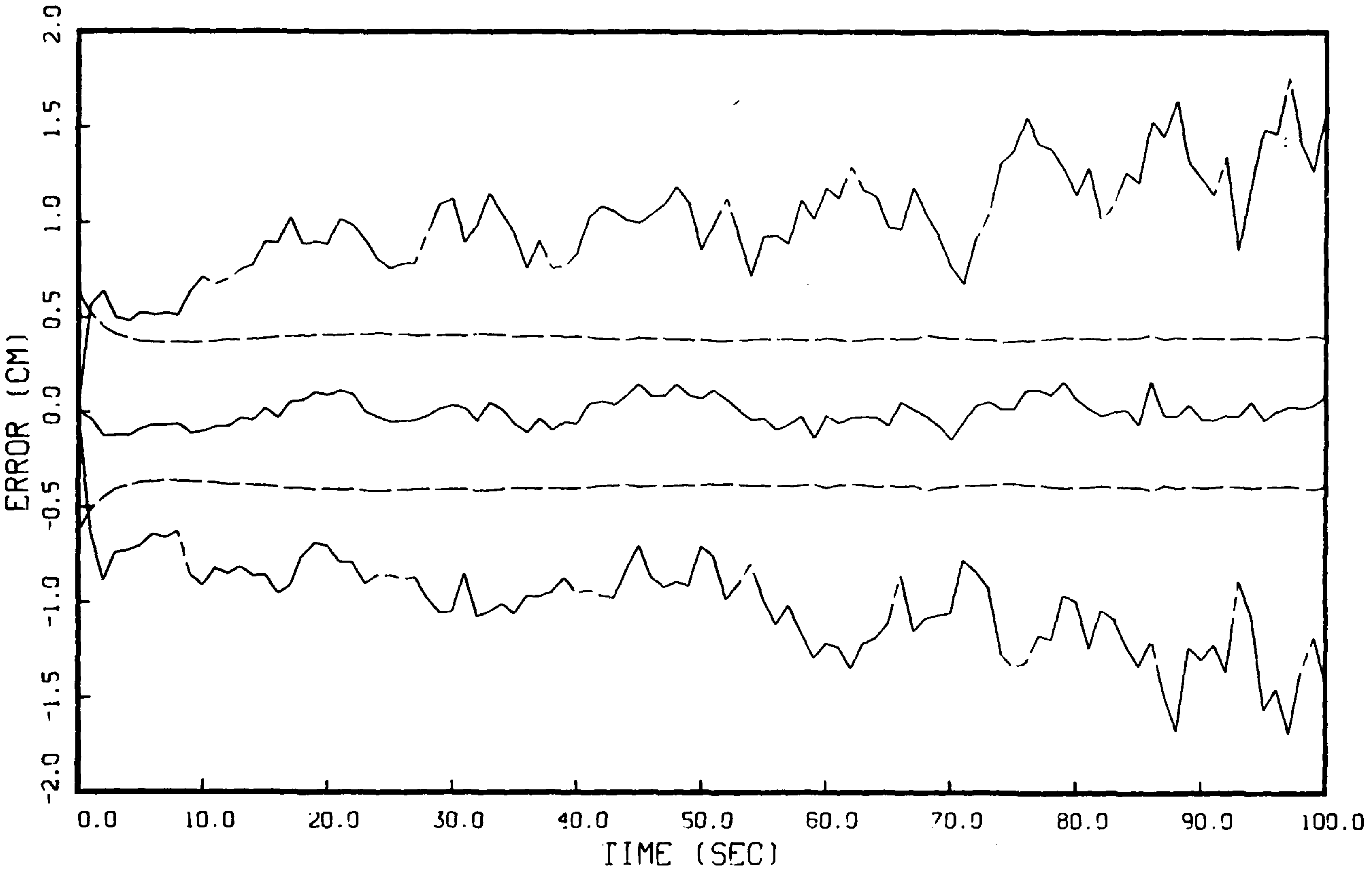


Figure 5.28 Closed Loop MMAE State Estimation Error
Mean +/- One Standard Deviation
True Tau = 20.0

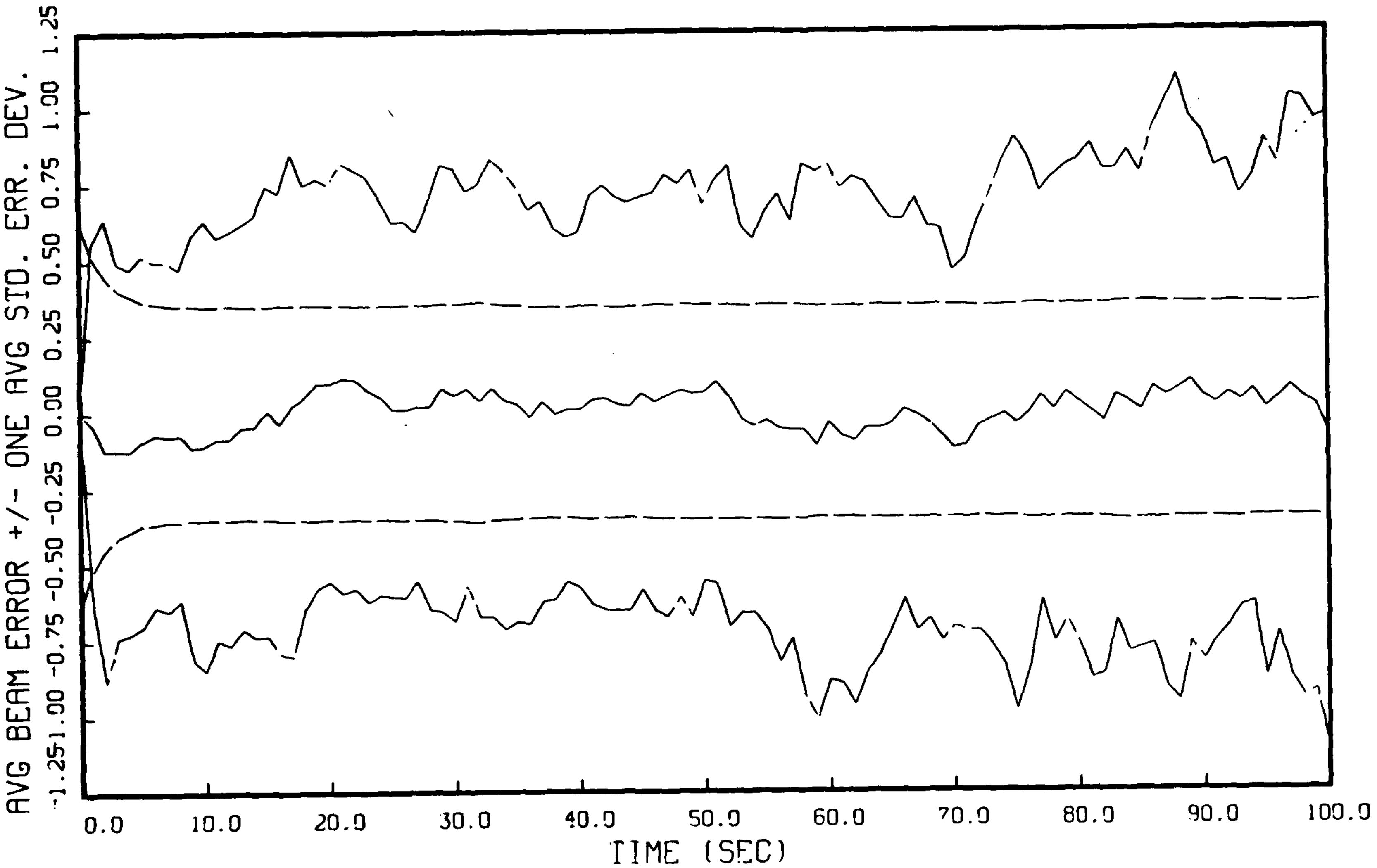


Figure 5.29 Closed Loop Meer #1 State Estimation Error
Filter Tau = 20.0 True Tau = 20.0

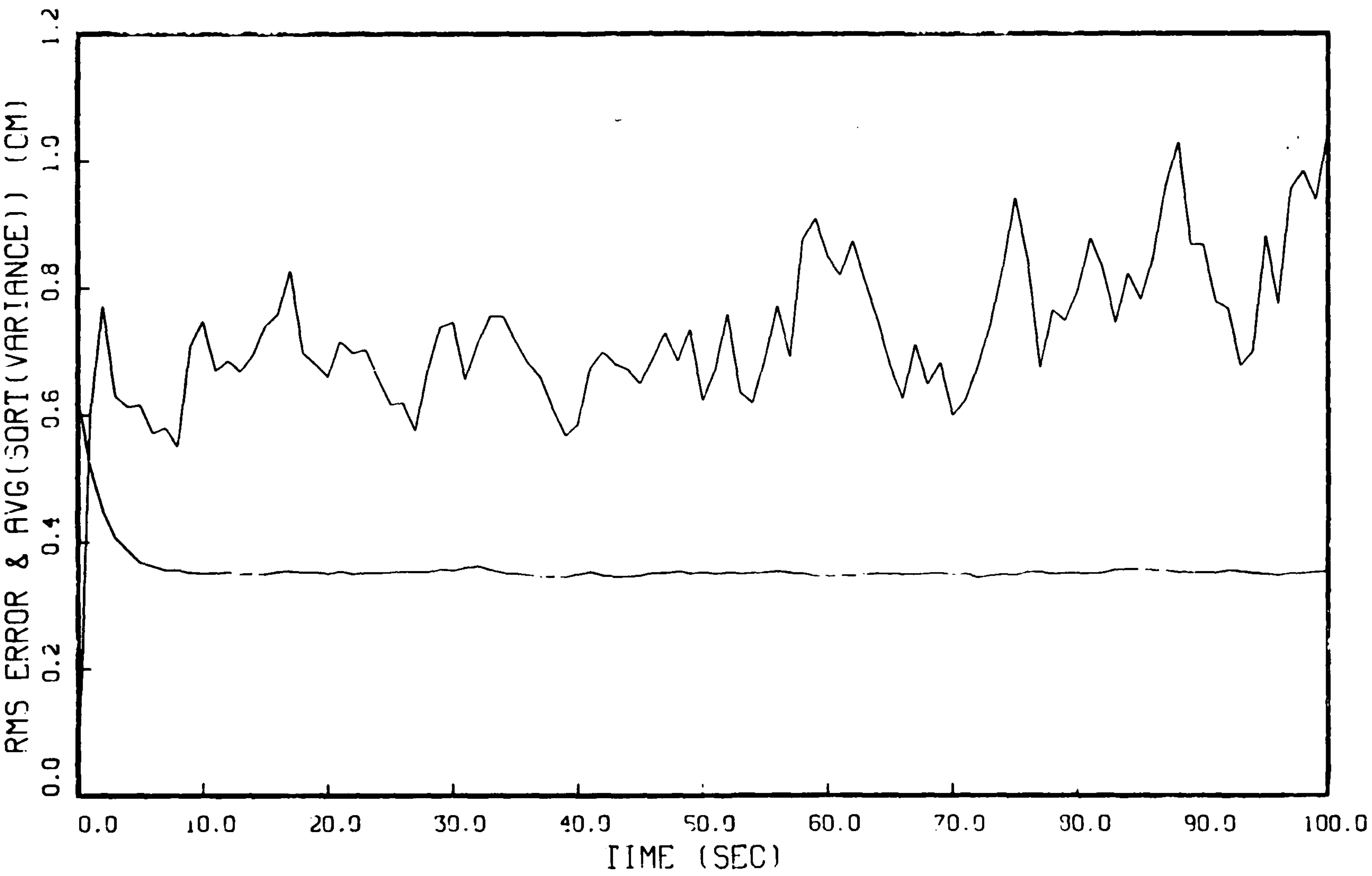


Figure 5.30 Closed Loop RMS Beam Error Meer #1
Filter Tau = 20.0 True Tau = 20.0

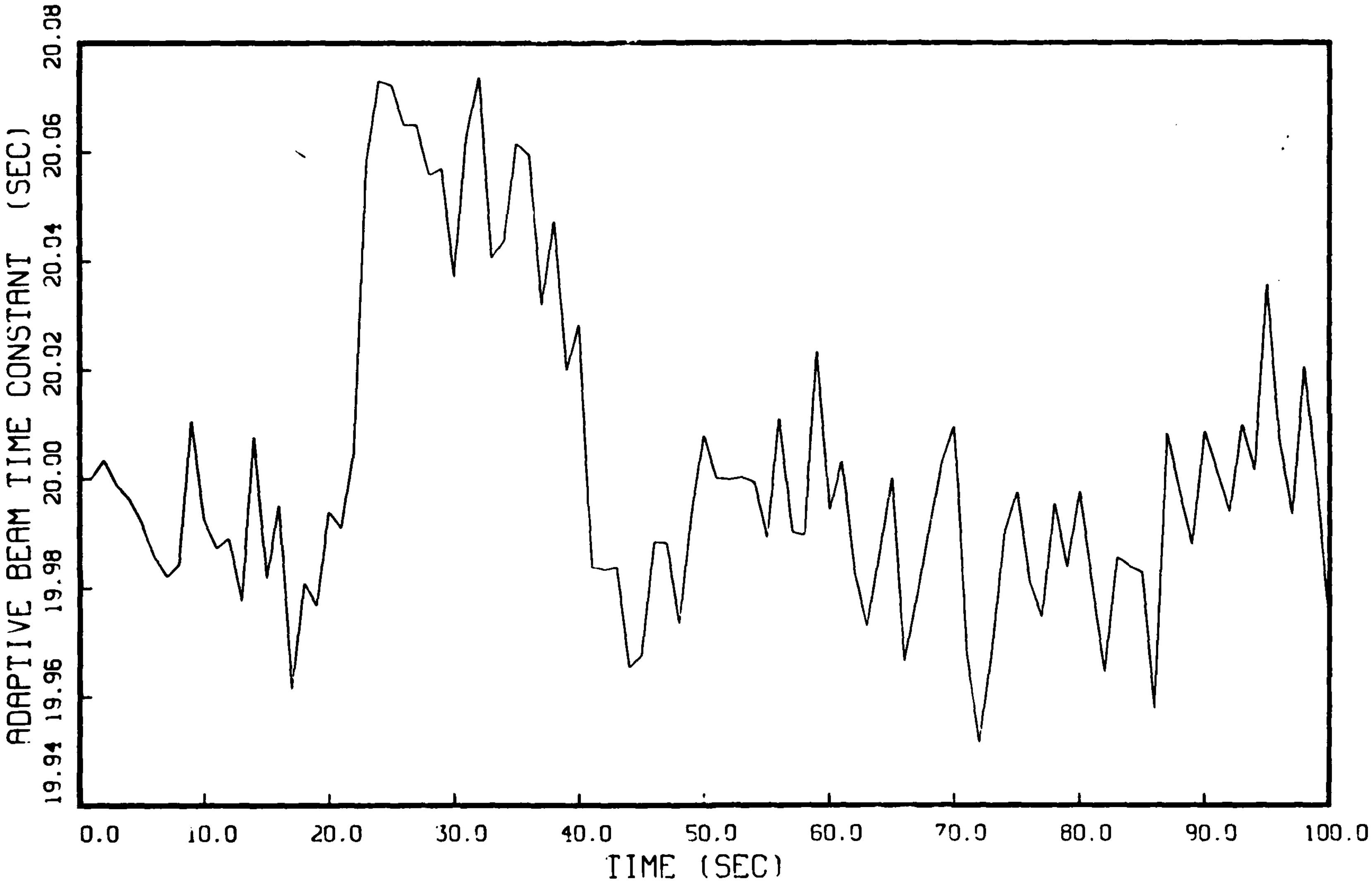


Figure 5.31 Closed Loop Mean Adaptive Beam Time Constant
True Tau = 20.0

we see the highest probability is correctly associated with the filter with beam time constant of 20. Figures 5.33 and 5.34 indicate that both of the other filters have equally low probability weights. Again, reasonably good performance. However, we might have expected the probability weights of the filters with the incorrect beam time constant to be driven to the lower bound of 0.0005. However, the occurrence of a noise event can result in shifts of the probability weights for the individual Meer filters.

To investigate the degree that the Meer filter is deweighting the measurements, a Meer filter was operated with the correct beam time constant, initial conditions matching the truth model, and no measurements. As would be expected, the actual error standard deviation envelope in Figure 5.35 closely matches the filter predicted standard deviation, which is the square root of steady state error variance given by Equation (5-1). Figure 5.36 is the performance of a single Meer filter with measurements provided. What is significant between these two figures is the similarity between actual error standard deviations and the steady state variance of the beam estimate. They are nearly the same in steady state. This, once again, indicates that the Meer filter is improperly deweighting measurements and is relying on its internal dynamics model. The reason for this is seen in Figure 2.4, the slopes of the signal rate function are rather steep. Thus, a small variation of the filter-estimated beam position from the

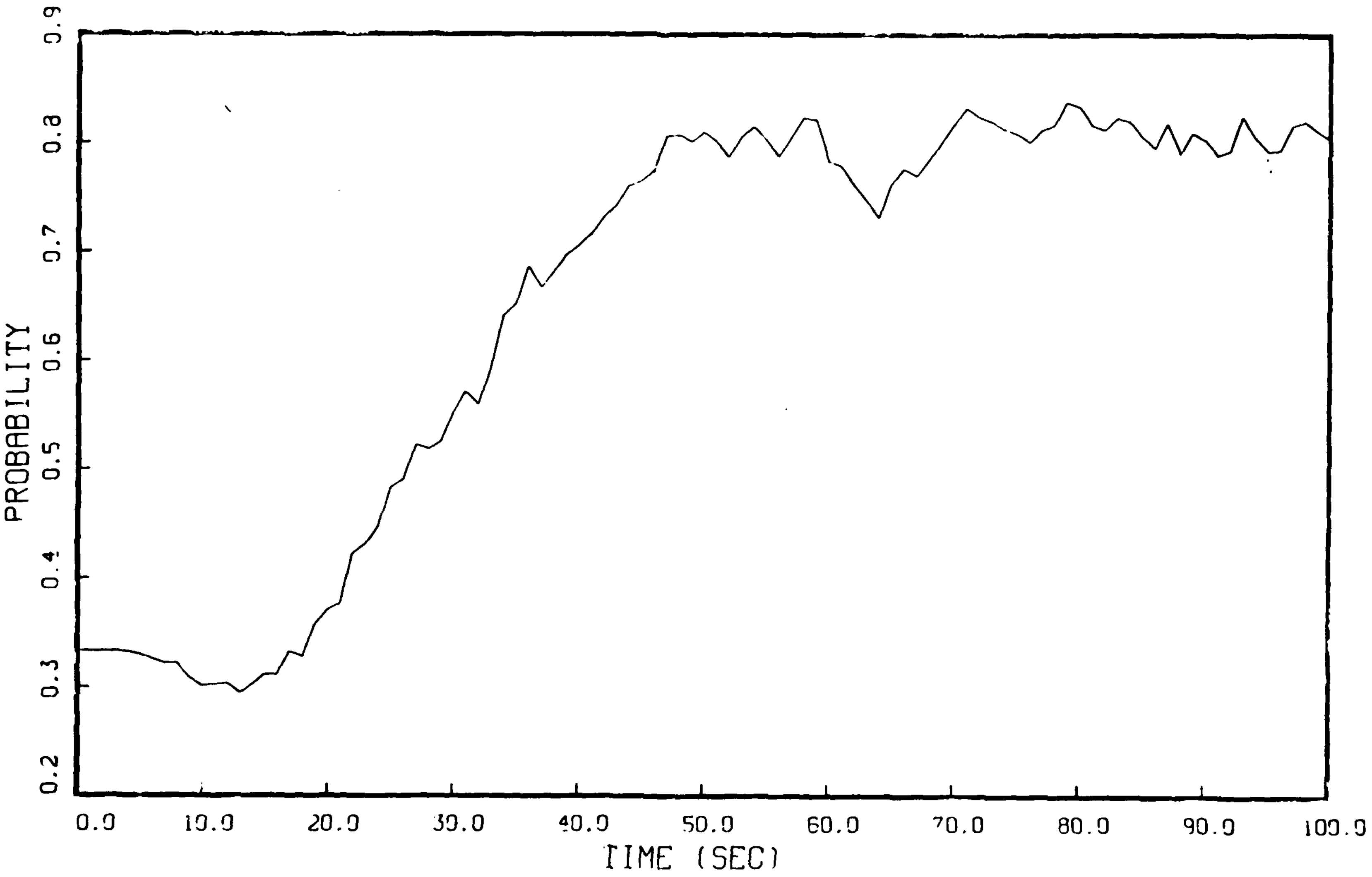


Figure 5.32 Closed Loop Mean Probability Meer Filter #1
Filter Tau =20.0 True Tau = 20.0

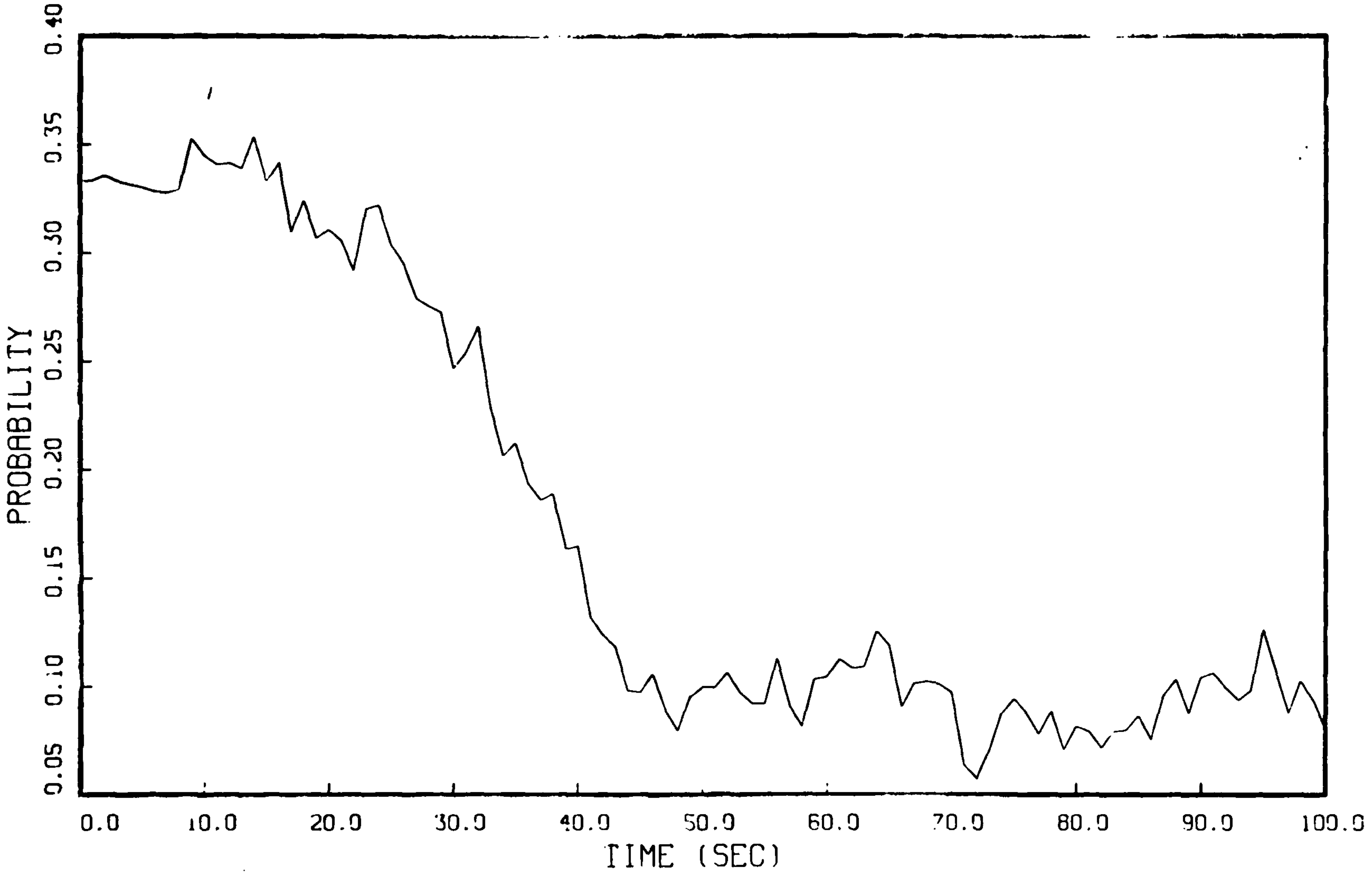


Figure 5.33 Closed Loop Mean Probability Meer Filter #2
Filter Tau =20.8 True Tau = 20.0

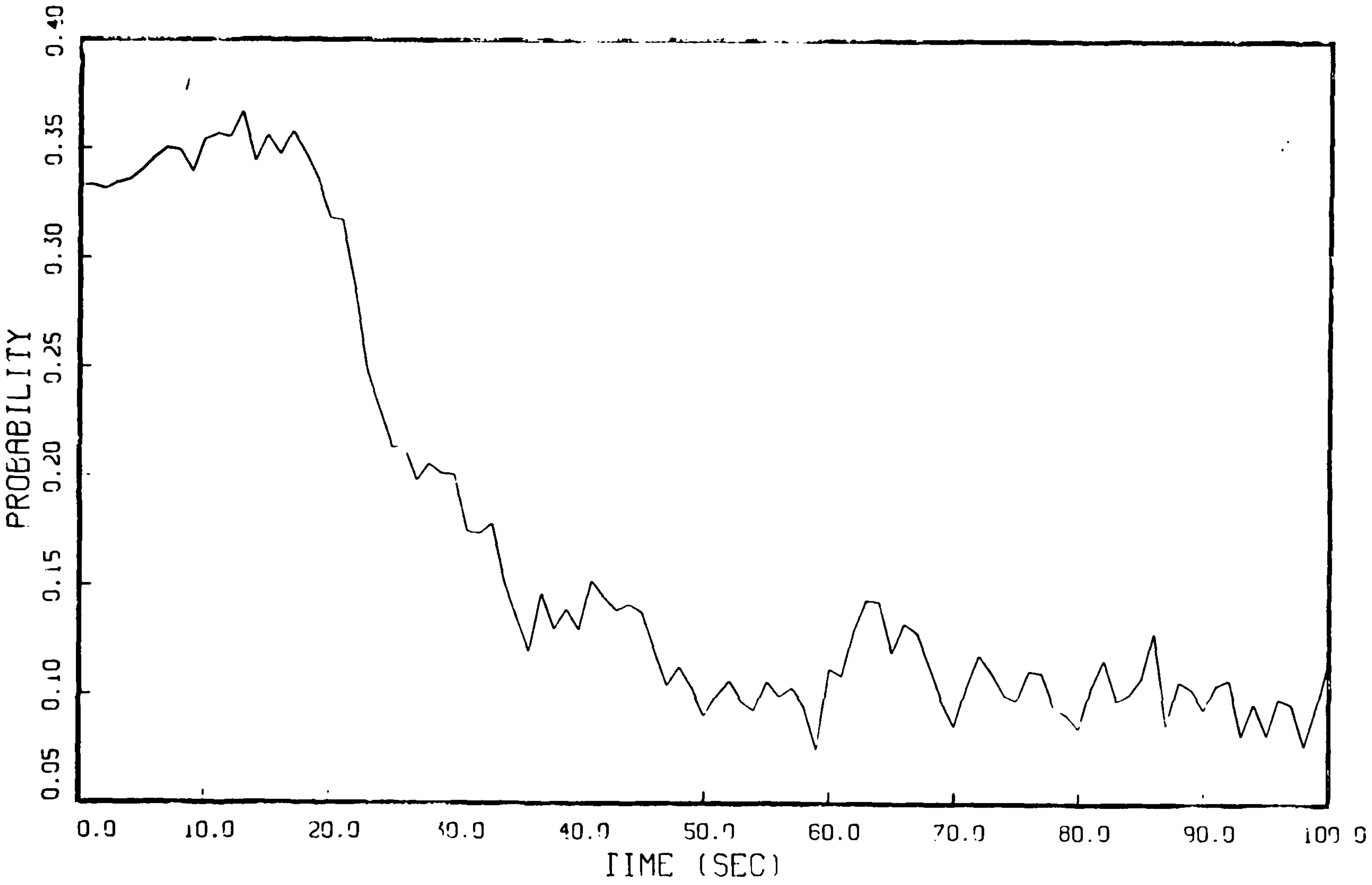


Figure 5.34 Closed Loop Mean Probability Meer Filter #3
Filter Tau =19.2 True Tau = 20.0

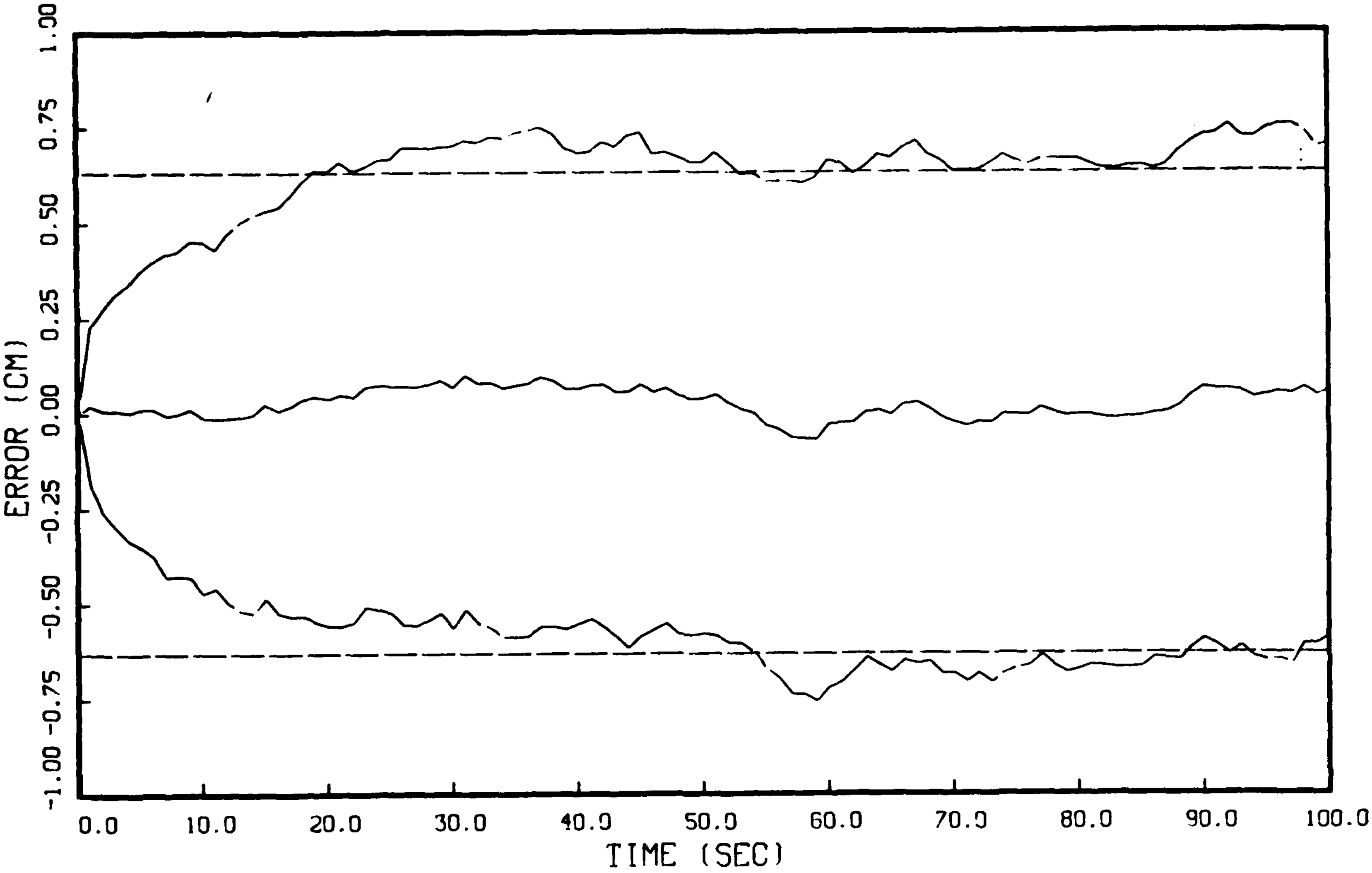


Figure 5.35 Meer Filter Without Measurements
Filter Tau =20.0 True Tau = 20.0

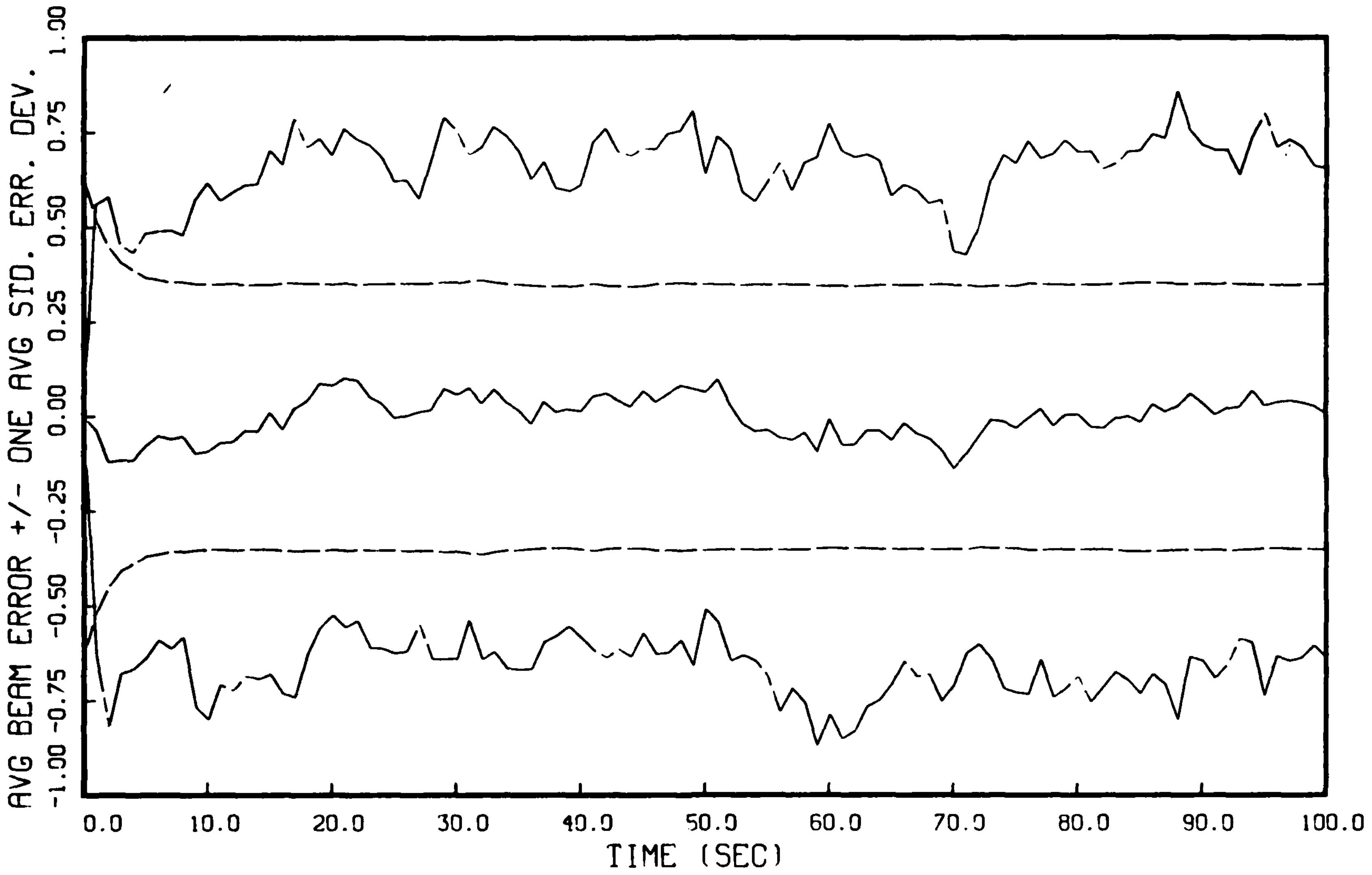


Figure 5.36 Meer Filter With Measurements
Filter Tau =20.0 True Tau = 20.0

actual beam position can result in a low estimate of the signal rate and the deweighting of elemental Snyder-Fishman filters that assumed the event to be a signal. The reason we see this in the closed loop case and not the open loop case is the fact that in open loop the truth model and the filter models will propagate toward each other if the sign of the true beam position is opposite that of the filter assumed position (all propagate toward zero), whereas in closed loop, the sign of the true beam position and the filter-assummed beam position will generally be the same as the beam is driven toward the target and generally away from zero. Again this is a result of operating with absolute states rather than relative states. Note in Figure 5.36, that the error standard deviation envelope is not growing. This demonstrates that a Meer Filter operating outside the MMAE Meer structure, with the correct beam time constant, and starting at the same initial conditions as the truth model, is at least stable. This is what we would expect. When compared to Figure 5.29 we see a possible indication that noise events are triggering the unnecessary reset of this filter while in the MMAE Meer structure.

5.6.4 Controller Baseline. To establish the controller baselines, the performance of the controller with full state feedback that was previously discussed and the performance of the same controller with estimates provided by a single Meer and a single Kalman filter are depicted in Figures 5.37

and 5.38. The Meer filter and Kalman filter both use the true parameter values. As expected, the true RMS error (Figure 5.37 lower line) for full-state feedback is very small. The upper line is the RMS error as calculated from the error between the MMAE Meer estimate and the target Kalman filter based on the true dynamics driving noise strength. In Figure 5.38 we see the effects of using filter estimates on tracking performance. Now the upper line is the true RMS error and the lower line is the RMS error based on filter estimates. As one would expect, the RMS error increases when full-state estimation is replaced with filter state estimates. Table 5.9 contains the RMS tracking errors for these two baselines. The RMS tracking error of the nine element MMAC is depicted in Figure 5.39. Here, the upper line is the actual tracking error whereas the lower line is the RMS tracking error as calculated from the MMAE Meer estimate and the Kalman filter estimate based on the true dynamics driving noise strength of 0.01. Note that the RMS error is increasing with time, indicating divergence. Such

Table 5.9 <u>Controller Baselines</u>

$R = .5\ cm^2$ $\quad q = .2\ cm/sec^{1/2}$ $\quad$ Depth = 1
True beam time constant = 20 seconds

controller type	RMS tracking error (cm)
Full-state Feedback	.212571
Filter Estimates	1.60719

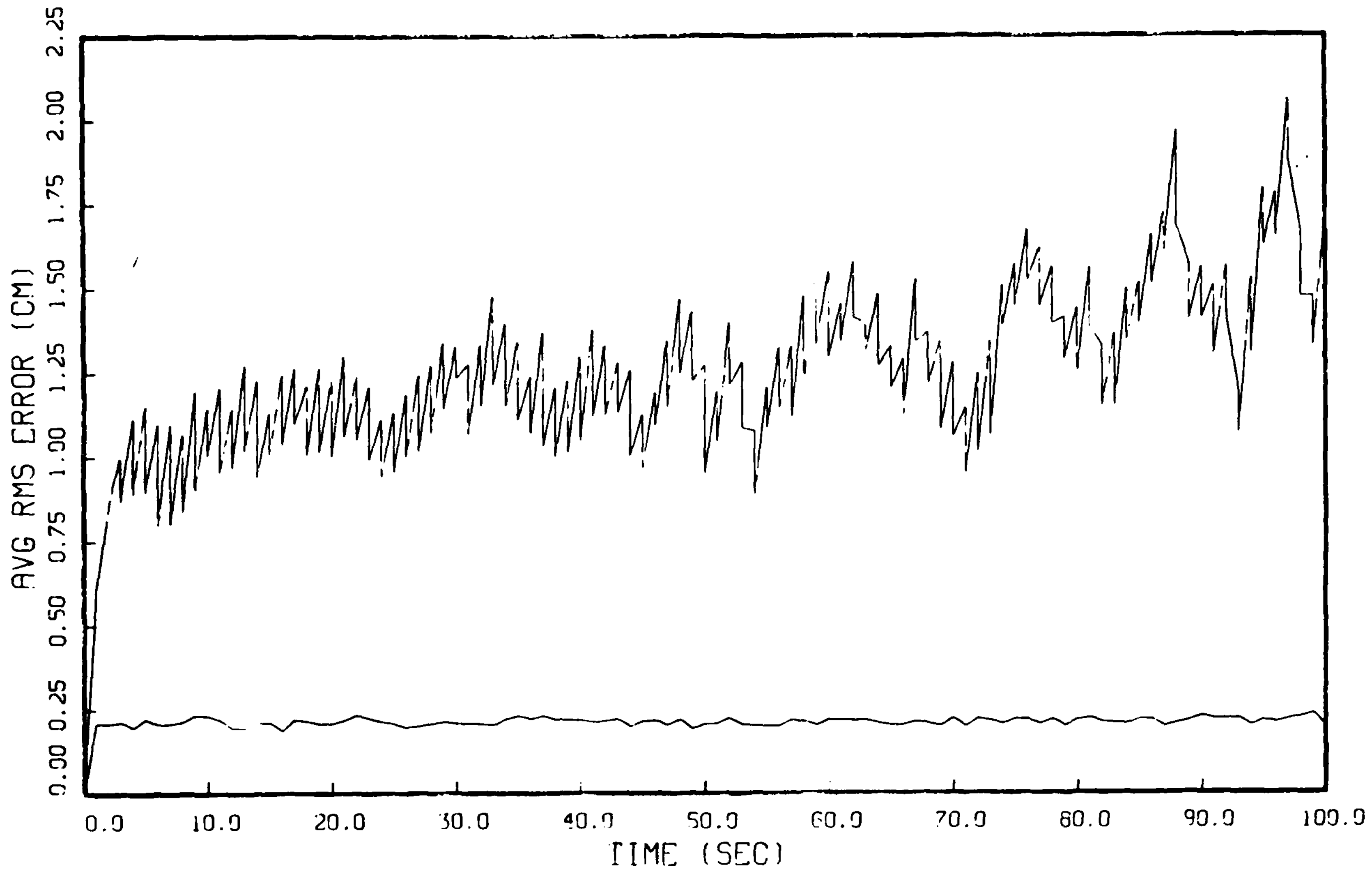


Figure 5.37 Controller RMS Tracking Error
Using Full-State Feedback

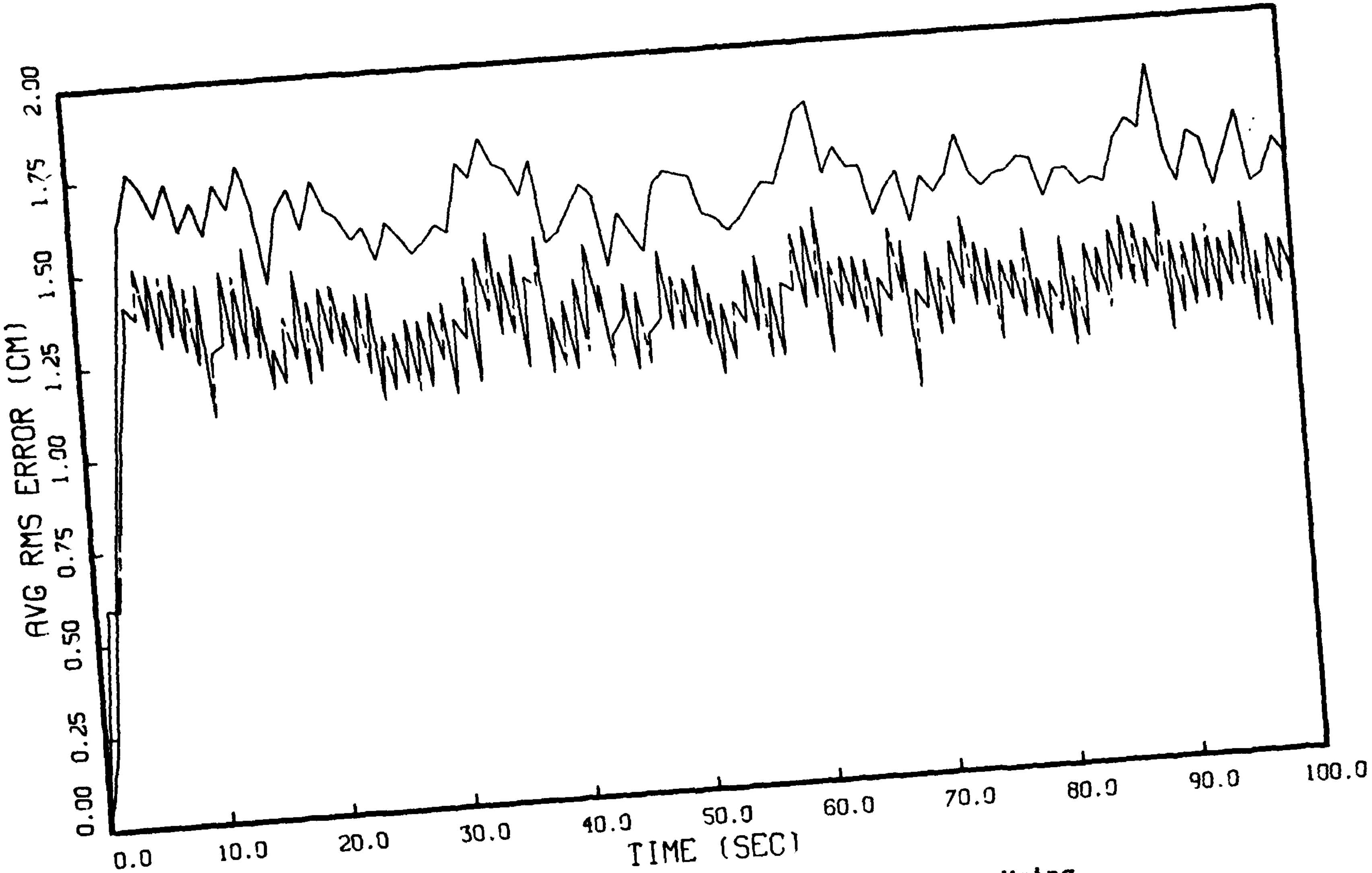


Figure 5.38 Controller RMS Tracking Error Using Single Filter Estimates

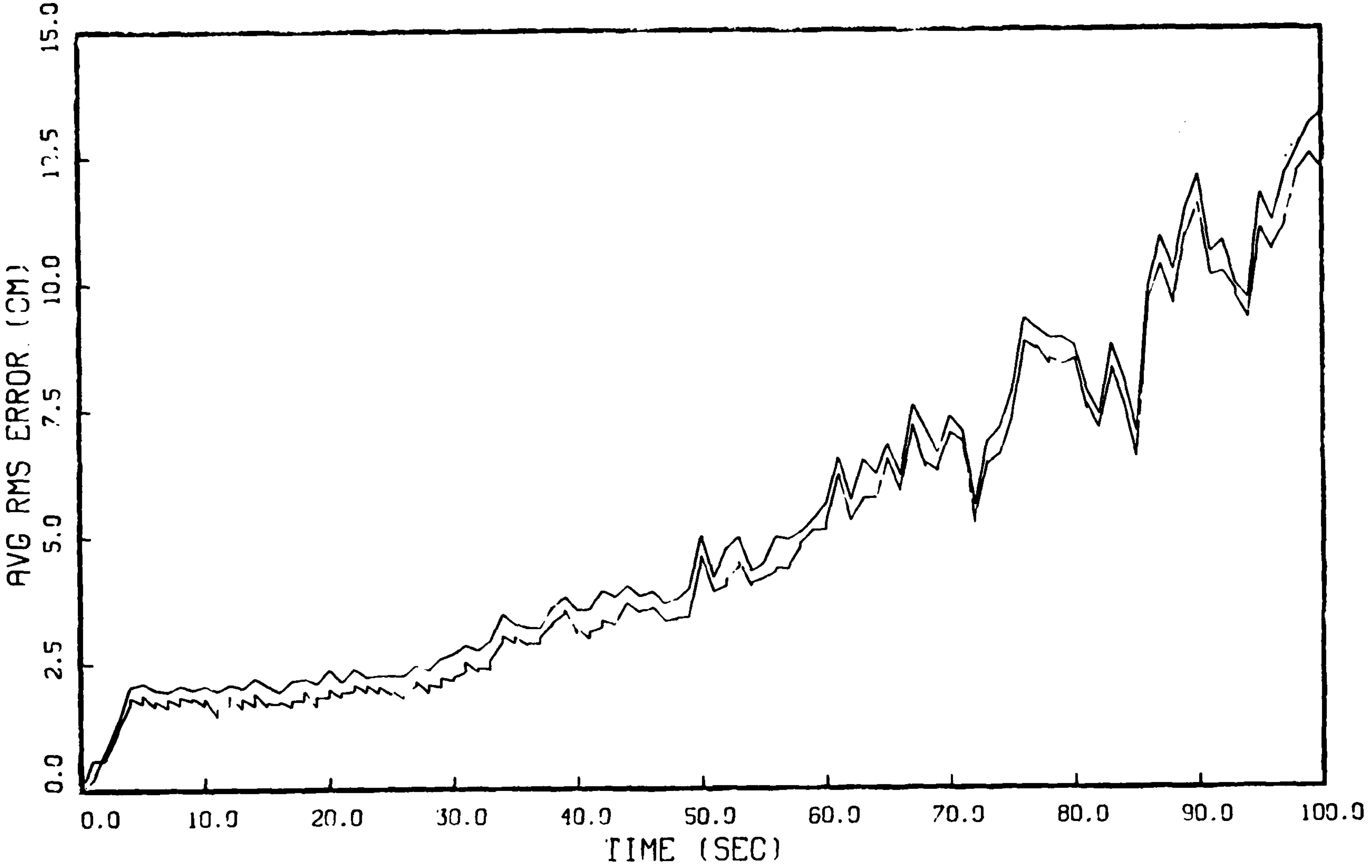


Figure 5.39 Nine Element MMAC RMS Tracking Error

poor performance may be due to the Kalman filter lower bound of .1. This bound results in poor elemental controllers receiving a greater probability weight than would normally occur.

5.7 Implicit Model Following Controller Performance

Programming errors coupled with time constraints prevented analysis of the Implicit Model Following controller design.

5.8 Summary

This chapter has presented the results of the analysis described in Chapter 4. Performance of the MMAE Meer filter with a wide variety of operating conditions was presented. While in open loop, the MMAE Meer filter is able to estimate the beam state with little difficulty. However, it is unable to estimate the beam time constant in open loop operation. The adaptive beam time contant estimate tends toward the largest beam time constant. This is attributed to the superior residual performance of a Meer filter with a large beam time constant. Open-loop performance of the MMAE Meer filter prompted an investigation of the closed-loop performance. This resulted in the examination of techniques to keep the beam over the array. One approach was to rotate the NPB toward the target. The results of this approach indicates that the measurement are being deweighted by the

Meer filter. The effect of the deweighting is an extended initial transient. However, steady state performance was good. In another approach, the detector is periodically repositioned under the beam. Although no long transient response is noted, there is filter divergence. To counter this characteristic, a likelihood function is used to detect divergence. When it diverges a Meer filter is reset to the MMAE Meer filter state estimate. Under these condition the state estimation performance is still divergent. However, there is improvement in the parameter estimation performance of the MMAE Meer filter. When this parameter is used to propagate the adaptive state estimate, the performance of the MMAE state estimation is superior to the elemental estimates. However, analysis indicates that again the measurements are being deweighted in the elemental Meer filters. Throughout the analyses, there is seen a dramatic difference in state estimation performance and parameter estimation performance between the open loop case and the closed loop case. This difference is also seen in the different methods used to track the target. This is attributed to the difference between using relative states and absolute states.

Two performance baselines are established for a controller designed around the the weighting matrices $X_{11} = 100$, $X_{55} = 10$, and $U = 1$ as used by Johnson. The first baseline is based on this controller receiving full-state feedback. The second baseline is based on this same controller receiving filter estimates from a single Meer filter and a single Kalman filter based on the true parameter values. The RMS tracking error for a nine-element MMAC is presented. This controller appeared to be more unstable than a three-element MMAC receiving adaptive beam state estimates from the MMAE Meer.

VI. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Eight analyses of MMAE Meer filter and a nine-element MMAC controller were completed. The MMAE Meer filter for this study was composed of 3 elemental Meer filters based on separate assumed beam time constants. The first five analyses studied MMAE Meer filter performance with an uncontrolled beam. The last three analyses examined the MMAE Meer filter performance with a controlled beam. As part of the closed loop analyses, baselines with the three-element MMAC used by Johnson [9] were established. Finally, the RMS tracking performance of the nine-element MMAC was presented. The first analysis indicates that there was little change in state estimation performance for a small mismodelling of the beam time constant for an uncontrolled beam. Results of the second analysis show that the MMAE Meer filter was unable to provide an accurate parameter estimate for an uncontrolled beam. Although there was no major computational savings realized in this simulation for operating at a Meer filter depth of one due to the computational load of other portions of the software, the third analysis demonstrated that a lower filter depth was desirable based on the RMS error performance for beam state estimation. As in the previous analyses, the fourth analysis indicates that parameter estimation performance for an uncontrolled beam was poor when the beam time constant was linearly or sinusoidually varying in time. The fifth analysis showed that state estimation performance was sensitive to very large mismodelling of the beam time constant for an uncontrolled beam. The results of the first five analyses indicate that an MMAE Meer filter was unnecessary for tracking an uncontrolled NPB, i.e. that a conventional Meer filter is sufficient.

However, the results of the last three analyses demonstrate that state estimation of the controlled beam is very sensitive to variations in the filter assumed beam time constant. In this case, the MMAE Meer filter is effective in providing a coarse estimate of the beam time constant, but there appears to be divergence in the beam state estimates. The analysis of this divergence indicates that the Meer filter is underweighting the measurement data. There was no evidence of a drastic loss of control; however, growth in RMS tracking error is evident. The final analysis demonstrated that the controller is sensitive to poor state estimates provided by the filters. The first baseline controller performance indicated that the controller design performed well with full-state feedback. Incorporation of filter estimates from filters based on the true parameters in the second baseline resulted in a substantial increase in RMS tracking errors. However, the RMS tracking errors did not increase with time, indicating at least stability at nominal conditions.

Analysis of the closed and open loop performances indicate there is a substantial improvement in state estimation performance when state values are maintained near zero (using relative states versus absolute states). The parameter estimation performance improved in the closed loop case as the propagation differences between the elemental Meer filter led to significant variations in residual performance. There is also an indication that divergence detection based on a likelihood function with a threshold of nine and N = 5 results in improper resetting of nondivergent filters.

No performance data was collected on a controller based on implicit model following techniques. However, there are conclusions that can be drawn from the development process. Implicit model following is not well-suited for problems with a first order Gauss-Markov plant, with one control input, and a single output. This reduces the model that can be followed to a single dimension (see Section 3.2.3.2). This does not offer a great deal of flexibility in closed loop pole selection. It must be noted that other techniques are also limited by the same constraint. Given a desirable closed loop pole based on design specifications for performance and robustness, this method does provide a straightforward approach to matching that pole.

6.2 Recommendations

These recommendations are divided into three parts. The first part concerns continued research with the beam estimator. The second concerns controller development. The last involves other avenues of research and improvement of the research tools.

Of foremost importance to the continued use of the Meer filter in beam control is the demonstration that the Meer filter can be tuned for closed loop operation. As indicated in Chapter V, the Meer filter apparently well tuned for open loop operations, appears to be poorly tuned for closed loop operation. Once tuning capability is demonstrated, it will be necessary to repeat sensitivity and robustness studies for variations in the beam parameters. The first step in attacking this tuning problem should be a sensitivity analysis of the Meer filter with respect to the beam dispersion R, with the detector properly located underneath the beam, during closed loop operations. Either of two methods can be considered for maintaining the beam over the detector. If rotation of the coordinate frame (using relative variables) is used, then the emphasis will be on tuning for transient response. If the detector is swept under the array (using absolute variables), then the emphasis will be on the steady state behavior. The goal of either approach will be to allow the Meer filter to make more effective use of measurement. However, if the second approach is selected, it is recommended that the detector be continuously repositioned under the array instead of periodically as done in this study. Properly locating the detector may necessitate modelling the detector motion and providing appropriate control to ensure all the signal events occur on the detector array. The particular values of R and square root of beam dynamics driving noise g used in this study may have contributed greatly to the apparent deweighting of measurements. The results of the sensitivity study may indicate a more appropriate value or range of values of R for subsequent study. The second step may involve the purposeful detuning of the Meer filter at open loop conditions to provide better performance for closed loop conditions. Increasing the square root of the dynamics driving noise g in the filter is the simplest method. This would boost the covariance and provide additional gain to the measurement. However, such a gain would be applied indiscriminately and may result in undesirable performance in noisy environments.

Although the MMAE Meer filter was effective in providing a coarse beam time constant estimate in the feedback mode, a much finer estimate may be possible with a finer discretization of the beam time constant. This would involve placing additional Meer filters within the MMAE Meer structure. This may also reduce the phase lag in estimating a sinusoidually varying beam time constant. Therefore it is suggested that performance of the MMAE Meer with additional elemental Meer filters be evaluated. If absolute variable are used then it also recommended that the number of elemental filters used in the MMAE Meer become a function of the beam state estimate. In addition, the state estimate used to reset divergent filters should be based on the nondivergent filters.

One aspect of the MMAE Meer filter that needs investigation is its performance under very noisy conditions. The probability weights used to determine the adaptive state estimate from the individual Meer filter state estimates are based on the measurement history or residual performance. Since noise events are a part of that residual performance, they may have a substantial impact on performance. When a noise event occurs it increase the probability of poorly performing filters. For low noise conditions, the subsequent signal events would cause the probabilities to fall for filters that are not accurately predicting the beam location. In a very noisy environment, it is conceivable that a sequence of noise events could occur that would result in a high probability weight being assigned to a filter that is actually providing a poor estimate of the beam location. Such a sequence of noise events could also result in the resetting of filters that are providing good estimates of the beam position, but have very high residuals with respect to the noise events. This could result in loss of beam tracking.

Another aspect of the MMAE Meer filter that needs investigation and perhaps some development, is the ability of the MMAE Meer filter to provide its own estimates of the maximum amplitude of the signal rate function, Γ, and the noise rate. Realistically, we will only be able to estimate the actual Γ from our assumed beam dispersion and a guess about the number of photons that would arrive over a given time interval. The Meer filter decision process may then be called upon to update this estimate based on its determination of the number of signal events that have occurred over a given period of time. Similarly, the Meer filter's determination of which events were noise events could be used to provide an estimate of the noise rate. Changes in laser and detector efficiencies provide the physical motivation to seek a real-time estimate of Γ and the noise rate for use by the Meer filter. Therefore, this estimation capability needs to be tested and perhaps developed.

Another area of estimator development that needs consideration is the likelihood function for divergence detection. As previously mentioned, the occurrence of successive noise events may incorrectly indicate divergence of a filter that is actual tracking the beam quite precisely. This filter would then be unnecessarily reset. However, it may be possible avoid the false divergence indication by varying the setting of the threshold and depth of the likelihood function based on the current estimate of Γ and the noise rate. The objective would be to provide adequate divergence checking without becoming susceptible to the noise. However, this must be weighed against the impact of adaptively estimating to many parameter. This could result in identifiability problems and possible instabilities.

Although little was accomplished in analysis of an Implicit Model Following controller design in this study, some recommendations are nevertheless made for further study. To satisfy sufficient conditions for a solution to the Riccati equation for controller gain synthesis, it was nescessary to shift two poles of the target model used for this synthesis. It is possible to avoid this modification to the target model used for gain synthesis by incorporating the target model into a Command Generator Tracker [18] developed as a precompensator feeding into a Proportional plus Integral controller for the beam.

In addition, no provision currently exists in the simulation code to isolate the impact of providing MMAE Meer estimates and Kalman filter estimates to the PI controller. Apparently, this code was removed after Zicker completed his study of the PG controller with the first-order target [32]. As targets, filters, and controllers become more sophisticated, the ability to isolate performance problems to a specific part of the system will become increasingly important. It is recommended that this capability be returned to the simulation. This isolation process can be divided into four phases. The first phase is full-state feedback. The second phase uses MMAE Meer filter feedback and target full-state feedback. This would allow isolation of problems strictly due to the use of MMAE Meer filter estimates. It may also provide insight into appropriate methods to recover robustness. Similarily, phase three would use full-state beam feedback and Kalman filter estimates to isolate and correct problems in this area. The final, fourth phase would use both MMAE Meer filter and Kalman filter estimates to determine what problems arise from the interaction of the estimates within the controller. This can also provide various baselines for controller performance.

As Johnson recommended [9:167], it is also recommended here that the number of runs necessary to provide sufficient convergence of sample statistics to true underlying statistics for controller evaluation should be rechecked. It is recommended that this be accomplished by embedding the equations used in SOFEPL [6:Sec 4] to provide the Monte Carlo statistics in a user defined subroutine in SOFE. Appropriate statistics to determine convergence of the sample statistics to underlying statistics could be calculated at the end of each run using the data from all the runs to that time in the Monte Carlo simulation. The number of runs necessary for a desired level of convergence could then be determined.

The last two recommendations concern the reduction of the SOFE computational load. With the recommendation to incorporate addition Meer filters into the MMAE Meer structure comes the problem of increasing the computer time required to perform each analysis. The first recommendation is to replace the differential equations used to propagate the truth and filter models with difference equations based state transition matrices. The linear models currently used do not require differential equations to provide precise propagation. If the next study desires or requires the use of the differential equations, then it is recommended that the differential equation of the dynamics driving noise strength be incorporated into the truth model as suggested in the SOFE manual [24:16]. This would allow the separate noise injection period to be eliminated. Instead, noise would be injected into the truth model at the time of Meer filter update and Kalman filter update only. Use of the differential equation would allow the appropriate discrete time noise to be determined despite the variable time interval. This method is not expected to reduce the required computer time as much as the first method, but as the current noise injection period is 0.2 seconds, it may reduce the number of calls to the integration routine by a factor of 3 by only integrating to each Poisson event time (average of one per second in this study) and to the Kalman filter update time (once per second).

Bibliography

1. Broussard, John R. and Paul W. Berry, "The Relationship Between Implicit Model Following and Eigenvalue Eigenvector Placement", IEEE Transactions on Automatic Control, AC-25 No 3, pp 591-593, June 1980.

2. Davenport, Wilbur B. Jr., Probability and Random Processes. New York: McGraw-Hill Book Company, 1970.

3. D'Azzo, John J. and Constantine H. Houpis, Linear Control System Analysis and Design (Second Edition). New York: McGraw-Hill Book Company, 1981.

4. Dixon, W. J., and F. J. Massey, Introduction to Statistical Analysis (Second Edition), McGraw Hill Book Company, New York, 1957.

5. Deming, W. E., Some Theory of Sampling, John Wiley and Sons, New York, 1950.

6. Feldman, R. E., and S. H. Musick, "SOFEPL: A Plotting Postprocesor for 'SOFE', User's Manual", AFWAL-TR-80-1109, Air Force Wright Aeronautical Laboratories, Wrigth-Patterson AFB, OH, November 1982.

7. Gilbert, Elmer G. "Conditions for Minimizing the Norm Sensitivity of Characteristic Roots," Conference on Information Sciences and Systems, Baltimore, Maryland, March 1978.

8. Jamerson, Lawrence C. Particle Beam Tracker for an Acclerating Target, MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, December 1985.

9. Johnson, Bruce A. Stochastic Adaptive Particle Beam Tracker Using Meer Filter Feedback, MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, December 1986.

10. Johnson, Bruce A., and Peter S. Maybeck. "Stochastic Adaptive Tracker Based on Noise-Corrupted Space-Time Measurement Process," unpublished paper of results obtained in the Masters thesis by Bruce A. Johnson.

11. Kreindler, E., and D. Rothschild. "Model-Following in Linear-Quadratic Optimization," AIAA Journal vol 7: pp 835-842. July 1976.

12. Kwakernaak, Huilbert, and Rapheal Sivan, *Linear Optimal Control Systems*, New York: John Wiley and Sons, Inc., 1972.

13. Lange, James J., and Howard E. Evans, "Elements of Remote Sensing," Course Notes for Physics 521, Space Surveillance" (Version B), Air Force Institute of Technology (AU), Wright-Patterson AFB OH, May 1985.

14. Lupo, James. Lecture notes from Physics 523 "Space Weapons", Air Force Institute of Technology, Summer Quarter 1987.

15. Maybeck, Peter S., William G. Miller and Jean M. Howey. "Robustness Enhancement for LQG Digital Flight Controller Design", IEEE Preceedings of the National Aerospace and Electronics Conference, Dayton, OH, May 84.

16. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 1*. New York: Academic Press, 1979.

17. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 2*. New York: Academic Press, 1982.

18. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Volume 3*. New York: Academic Press, 1982.

19. Maybeck, Peter S., and William L. Zicker, "MMAE-Based Control with Space-Time Point Process Observations," *IEEE Transactions on Aerospace and Electronic Systems* AES-21: pp. 292-230 (May 1985)

20. McKee, Leslie L. Staff Notes for Physics 7.54, The Physics of Charged Particle Beams, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, August 1986.

21. Meer, David E. *Multiple Model Adaptive Estimation for Space-Time Point Process Observations*, PhD Dissertation, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, September, 1982.

22. Miller, W. G., "Robustness Multivariable Controller Designing Via Implicit Model-Following Methods," M.S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1983.

23. Moose, William J. *A Proportional-Plus-Integral Controller for a Particle Beam Weapon*, MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, December 1984.

24. Musick, S. H., "SOFE: A Generalized Digital Simulation for Optimal Filter Evauation, User's Manual", AFWAL-TR-80-1108, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, OH, October 1980.

25. Parmentola, John, and Kosta Tsipis "Particle-Beam Weapons", Scientific American, Vol 240 Number 4, pp. 54-65 (April 1979).

26. Rohringer, G. Particle Beam Diagnostics by Resonant Scattering, 1 June 1977-30 November 1977. Contract DASG60-77-C-0120. General Research Corporation, Santa Barbara, CA, December 1977.

27. Santiago, J. Fundamental Limitation of Optical Trackers, Masters's Thesis, AFIT, Wright-Patterson AFB, OH, December 1978.

28. Snyder, Donald Lee. Random Point Processes. New York: John Wiley and Sons. Inc., 1975.

29. Snyder, Donald L., and Philip M. Fishman, "How to Track a Swarm of Fireflies by Observing Their Flashes," IEEE Transations on Information Theory, IT-21: 692-695 (Nov 1975).

30. Upadhyay, Triveni N., and Jerry L. Weiss, "Applications of Filtering Techniques to Problems in Communication-Navigation Systems", Control and Dynamic Systems, Vol 21 pp 337-418. Orlando, Florida: Academic Press Inc., 1984.

31. Weiss, J. L., T. N. Upadhyay, and R. Tenney. "Finite Computable Filters for Linear Systems Subject to Time Varying Model Uncertainty," in Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON) May 1983 pp. 349-355; of results obtained in a Masters Thesis by Jerold L. Weiss, "A Comparison of Finite Filtering Methods for Status Directed Processes," Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1983.

32. Zicker, William L. Pointing and Tracking of Particle Beams, MS Thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, December 1983.

33. Houpis, Constantine H., and Gary B. Lamont. Digital Control Systems. New York: McGraw-Hill Book Company, 1985.

Vita

Captain Louis Jasper Harambasic Jr. was born on 10 June 1958 in Phoenix, Arizona. He graduated form high school in Prescott, Arizona, in 1976 and attended Arizona State University from which he received the degree of Bachelor of Science in Engineering in May 1980. Upon graduation, he received a commission in the USAF through the ROTC program. His first assignment was to the 2nd Communications Squadron, Buckley ANGB, CO., as a Satellite Systems Engineer. He entered the School of Engineering, Air Force Institute of Technology, in May 1986.

Permanent address: H.C. 68 Box 881
Kirkland, Arizona
86332

A194622

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GE/ENG/87D-23

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
Louis Jasper Harambasic Jr., B.S.E., Captain USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1987 December

15. PAGE COUNT
235

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	03	
19	12	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Neutral Particle Beam, Beam Handling and Control, Monte Carlo Method, Stochastic Processes, (cont)

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: Tracking and Control of a Neutral Particle Beam
Using Multiple Model Adaptive Meer Filter

Thesis Chairman: Peter S. Maybeck, PhD

Approved for public release: IAW AFR 190-1.
Lynn Wolaver 21 May 88
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH [illegible]

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Peter S. Maybeck

22b. TELEPHONE *(Include Area Code)*
(513) 255-3576

22c. OFFICE SYMBOL
AFIT/ENG

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

Block 19 cont.

Abstract

The purpose is to point the centroid of a Neutral Particle Beam (NPB) at an intended target. A Multiple Model Adaptive Estimator using elemental Meer Filters is used to estimate the centroid of a NPB model as a one-dimensional first-order Gauss-Markov position process. The MMAE Meer Filter is also used to estimate the beam time constant. "Merge Method" of filter pruning is used to limit the size of the elemental Meer filters. A bank of three Kalman filters are used to estimate the states of the target which has a variable dynamics driving noise strength. The target is modelled as a third-order Gauss-Markov position process. A Multiple Model Adaptive Controller is designed using LQG methods, and true states are replaced by their best estimates by invoking the principle of assumed certainty equivalance. MMAE Meer Filter performance analysis is performed for an uncontrolled beam and for a controlled beam. Controller baselines are established.

Block 18 cont.

Error Estimation, Bayes Theorem, Covariance, Distribution curves, Gaussian Noise, Markov process, Maximum likelihood estimation, Multiple Model Adaptive Control, Multiple Model Adaptive Estimation, Poisson density function, Pseudorandom systems,

END
DATE
FILMED
8-88
DTIC